Fine-grained Arabic Named Entity Recognition

by

Fahd Saleh S Alotaibi

A THESIS SUBMITTED TO THE UNIVERSITY OF BIRMINGHAM FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
The University of Birmingham
JANUARY 2015

Abstract

Named Entity Recognition (NER) is a Natural Language Processing (NLP) task, which aims to extract useful information from unstructured textual data by detecting and classifying Named Entity (NE) phrases into predefined semantic classes. This thesis addresses the problem of fine-grained NER for Arabic, which poses unique linguistic challenges to NER; such as the absence of capitalisation and short vowels, the complex morphology, and the highly inflection process.

Instead of classifying the detected NE phrases into small sets of classes (i.e. coarse-grained ranged from 3 to 10); we target a broader range (i.e. 50 fine-grained classes 'hierarchal-based of two levels') to increase the depth of the semantic knowledge extracted. This has increased the number of classes, complicating the task, when compared with traditional (coarse-grained) NER, because of the increase in the number of semantic classes and the decrease in semantic differences between fine-grained classes. Fine-grained NER is advantageous in various NLP tasks, including Information Extraction, Ontology Construction and Populations, and Question Answering among many others.

Our approach to developing fine-grained NER relies on two different supervised Machine Learning (ML) technologies (i.e. Maximum Entropy 'ME' and Conditional Random Fields 'CRF'), which require annotated (i.e. labelled) training data (i.e. a corpus) in order to learn by extracting informative features. Therefore, the development of such resources comprises one of the thesis contributions. We develop a methodology which exploit the richness of Arabic Wikipedia (AW) in order to create a scalable fine-grained lexical resource (gazetteer) and a corpus automatically. Moreover, two gold-standard cre-

ated corpora from different genres were also developed to perform comparable evaluation. The thesis also developed a new approach to feature representation by relying on the dependency structure of the sentence to overcome the limitation of traditional window-based (i.e. n-gram) representation. Furthermore, by exploiting the richness of unannotated textual data to extract global informative features using word-level clustering technique was also achieved. Each contribution was evaluated via controlled experiment and reported using three commonly applied metrics, i.e. precision, recall and harmonic F-measure.

Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful

I thank Allah for granting me the patience, health, guidance and determination to complete this thesis successfully.

I would like to express my special appreciation and thanks to my supervisor Dr. Mark Lee; you have been a tremendous supervisor for me. I would like to thank you for encouraging me in my research and for allowing me to grow as a research scientist. Your advice on both research and on my career has been priceless.

I would also like to thank my thesis group members, Dr. Peter Hancox and Dr. Behzad Bordbar for serving as committee members despite times of difficulty and hardship. I also want to thank you for allowing my defence be an enjoyable experience, and for your brilliant comments and suggestions, many thanks to you.

I extend my gratitude the Saudi Arabian Cultural Bureau in London for financially supporting me to pursue my PhD, and for making this long journey possible.

A special thanks go to my family. Words cannot express how grateful I am to my father (who died in January 2014), my mother, and my sister Mona (who died in December 2012), and for all of the sacrifices my brothers and sisters have made on my behalf. My deepest thanks go to my little family, my wife (Nawal), son (Yousef), and daughters (Leen, Jood and Lulua). Your prayer for me have sustained me thus far.

I would finally like to acknowledge all the friends who supported me in my writing, and incentivised me to strive towards my goal.

Contents

List of Figures

List of Tables

Arabic transliteration scheme

In this thesis, we relied on the transliteration scheme provided by Habash et al. (2007). Therefore, throughout this thesis and where appropriate, Arabic words are represented in three variants: (Arabic word /HSB transliteration scheme (Habash et al., 2007)/ 'English translation')[1].

Table 1: The used Arabic letters transliteration scheme

Arabic letter	Transliteration	Arabic letter	Transliteration	Arabic letter	Transliteration
ء	'	ذ	∂	ق	q
أ	O	ر	r	ك	k
إ	I	ز	z	ل	l
آ	$\overline{A}$	س	s	م	m
ا	A	ش	$\check{s}$	ن	n
ب	b	ص	S	ه	h
ت	t	ض	D	و	w
ث	θ	ط	T	ي	y
ج	j	ظ	$\check{D}$	ى	$\acute{y}$
ح	H	ع	ʕ	ئ	$\widehat{y}$
خ	x	غ	γ	ؤ	W
د	d	ف	f	ة	$\hbar$

Table 2: The used Arabic diacritics transliteration scheme

Arabic diacritic	Transliteration	Arabic diacritic	Transliteration	Arabic diacritic	Transliteration
◌َ	a	◌ِّ	$\tilde{i}$	◌ٓ	.
◌ِ	i	◌ُّ	$\tilde{u}$	◌ً	$\tilde{a}$
◌ُ	u	◌ّ	$\sim$	◌ـ	-

[1]In some cases, when presenting letters, prefixes, suffixes and diacritics, we only use (Arabic word /HSB transliteration scheme/) without English translation

The Arabic Parts of Speech Tagset

For consistency and as appropriate, we rely on the Reduced Tag Set (RTS) developed by Ann Bies and Dan Bikel to present the part of speech tags throughout this thesis (a full description of RTS is presented in (Habash, 2010, p.80)).

Table 3: Full description of the Reduced Tag Set (RTS)

Category	Tag	Usage
Nominals.Nouns	NN	Singular common noun or abbreviation
	NNS	Plural/dual common noun
	NNP	Singular proper noun
	NNPS	Plural/dual proper noun
Nominals.Pronouns	PRP	Personal pronoun
	PRP$	Possessive personal pronoun
	WP	Relative pronoun
Nominals.Other	JJ	Adjective
	RB	Adverb
	WRB	Relative adverb
	CD	Cardinal number
	FW	Foreign word
Particles	CC	Coordinating conjunction
	DT	Determiner
	RP	Particle
	IN	Preposition
Verbs	VBP	Active imperfect verb
	VBN	Passive imperfect/perfect verb
	VBD	Active perfect verb
	VB	Imperative verb
Other	UH	Interjection
	PUNC	Punctuation
	NUMERIC_COMMA	The letter used as a comma
	NO_FUNC	Unanalysed word

List of Abbreviations and Acronyms

Table 4: List of Abbreviations and Acronyms

Abbreviations	**Full form**
(ACE)	Automatic Content Extraction
(AI)	Artificial Intelligence
(ANN)	Artificial Neural Networks
(ATB)	Arabic Tree Bank
(AW)	Arabic Wikipedia
(BN)	Broadcast News
(BPC)	Base Phrase Chunk
(CA)	Classical Arabic
(CATiB)	The Columbia Arabic Treebank
(CLEF)	Conference and Labs for the Evaluation Forums
(CoNLL)	The Conference on Computational Natural Language Learning
(CRF)	Conditional Random Fields
(DT)	Decision Tree
(ELF)	Enhanced Language-dependent Features

Continued on next page

Table 4 – continued from previous page

Abbreviations	Full form
(ERTS)	The Extended Reduced Tag Set
(FAC)	Facility
(FF)	Filtered Features
(GPE)	Geographic and politic
(IDF)	Idafa, i.e. two nouns
(IE)	Information Extraction
(LF)	Language-dependent Features
(LOC)	Location
(LR)	Logistic Regression
(MADA)	Morphology Analysis and Disambiguation for Arabic
(MDA)	Mention Detection Algorithm
(ME)	Maximum Entropy
(ML)	Machine Learning
(MNB)	Multinomial Nave Bayes
(MSA)	Modern Standard Arabic
(MUC)	Message Understanding Conference
(MUC-6)	The Sixth Message Understanding Conference
(NB)	Nave Bayes
(NE)	Named Entity
(NEC)	Named Entity Classification
(NED)	Named Entity Detection
(NER)	Named Entity Recognition
(NLP)	Natural Language Processing
(NLTK)	Natural Language Tool Kit

Continued on next page

Table 4 – continued from previous page

Abbreviations	Full form
(NN)	Neural Network
(NW)	Newswire
(OBJ)	Object
(ORG)	Organisation
(OSV)	Object-subject-verb
(OVS)	Object-verb-subject
(PADT)	The Prague Arabic Dependency Treebank
(PATB)	The Penn Arabic Treebank
(PER)	Person
(POS)	Part of Speech
(QA)	Question Answering
(RE)	Relation Extraction
(SBJ)	Subject
(SF)	Simple Features
(SP)	Structured Perceptrons
(SVM)	Support Vector Machine
(SVO)	Subject-verb-object
(TF)	Term Frequency
(TF-IDF)	Term Frequency-Inverse Document Frequency
(TMZ)	Tamyiz
(TP)	Term Presence
(VEH)	Vehicles
(VOS)	Verb-object-subject
(VSO)	Verb-subject-object

Continued on next page

Table 4 – continued from previous page

Abbreviations	**Full form**
(WB)	Weblogs
(WEA)	Weapons
(YamCha)	Yet Another Multipurpose CHunk Annotator
(YASMET)	Yet Another Small MaxEnt Toolkit

Part I

INTRODUCTION

Chapter 1

Introduction

1.1 Overview and Motivation

Living in today's modern and complex world, we are surrounded by abundant information in different forms, including texts, images, audio data and videos. Media, and knowledge centres, such as the newswire agencies and libraries, are continually producing more and more data. Online encyclopaedias and microblogging, i.e. social sites, are also contributing to the available knowledge via a wide spectrum of users with different attitudes. The common denominator with all these resources, is, that when presented in an unstructured way, they complicate the user's ability to locate valuable information.

In the case of the integration of Artificial Intelligence (AI) technologies, the field of Information Extraction (IE) is dedicated to assisting, extracting and then structuring data in a way that makes it useful for end users. Among valuable data is Named Entity (NE) mentions, referring to existing real life objects, which belong to semantic categories such as person, organisation and location. To meet the need for data extraction, Named Entity Recognition (NER), as a subfield of Information Extraction (IE), has been established. It incorporates two sub-tasks; Named Entity Detection (NED) and Named Entity Classification (NEC) (Nadeau and Sekine, 2007), although the majority of researchers tackle these in one step, such as Benajiba et al. (2007). The first describes the ability to detect the boundary of NEs from context, requiring correct identification of start and the end tokens for each NE phrase. The second task is to classify the entity detected into one of several pre-defined semantic classes, e.g. person, organisation or

location. NER is not however a straightforward task. Even for languages that provide significant clues such as capitalisation, e.g. English, complex NEs, such as the titles of books or movies can only be distinguished with significant effort (Downey et al., 2007; Dale and Mazur, 2007). This problem becomes more apparent when the language concerned includes no orthographical signs, e.g. Arabic.

Arabic, which is the target language of this research, has no character level indicators to assist in differentiation between proper nouns and common nouns or verbs. Moreover, Modern Standard Arabic (MSA) is written from right to left with the absence of diacritics, i.e. vowels. It is considered a complex morphological language, with high inflection, due to the process of the agglutination of the prefixes and suffixes to the stem. These challenges, combined with many others, have led researchers to consider different sources of additional information that are either language-dependent or independent, such as Benajiba et al. (2008a); AbdelRahman et al. (2010).

Development of NER systems can be achieved either by relying on a list of handcrafted rules, or by recruiting statistical machine learning algorithms and extracting a set of informative features to be recruited into the learning algorithm (Nadeau and Sekine, 2007; Shaalan, 2013). Each approach has its own benefits and limitations. Handcrafted rules require a linguistic expert from the target domain to extract useful rules. How these generalised rules can be applied to different domains then becomes a critical issue. Therefore, amendments to the rules or the addition of new rules are a necessity. Reliance on machine learning, on the other hand, has attracted many researchers due to the generalised nature of the approach and the lesser requirement for linguistic knowledge. This approach requires an annotated, i.e. a labelled, dataset to learn from. In addition to the dataset, two issues are crucial to investigate. The first issue is the probabilistic model, i.e. the learning algorithm. The second is the set of features extracted from the data for use by the learning algorithm.

Despite the fact that there have been a number of attempts to address recognition of Arabic named entities; such as Benajiba et al. (2009a); Darwish (2013); Morsi and Rafea (2013), all the works (to the best of the author's knowledge) have been restricted to the newswire domain, and presentation of a very limited number of traditional classes. Very limited semantic classes or types (i.e. called coarse-grained) are not enough nowadays to cover the need of the users' queries

and applications, such as Question Answering.

Therefore, in this thesis, we differentiate between coarse-grained and fine-grained NER in which the former tackles a single level of small predefined classes (such as Person, Organisation and Location) whilst the latter involves hierarchal representation of the classes with at least two levels. Figure 1.1 and Figure 1.2 present examples of coarse- and fine-grained tagset respectively. Fine-grained NER is considered much harder to develop than classical NER, because of the increase in the number of semantic classes and the decrease in the semantic differences between classes (Li et al., 2012). Development of a fine-grained NER applied to languages other than Arabic has begun to attract researchers (such as English (Li et al., 2012) and Dutch (Desmet and Hoste, 2014)). Fine-grained NER is ideal for use in evolving fields, such as Ontology Construction and Populations, and Question Answering (Fleischman and Hovy, 2002; Mollá et al., 2006; Lee et al., 2006).

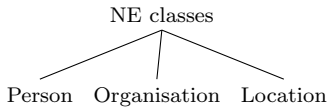

Figure 1.1: An example of coarse-grained single-level tagset

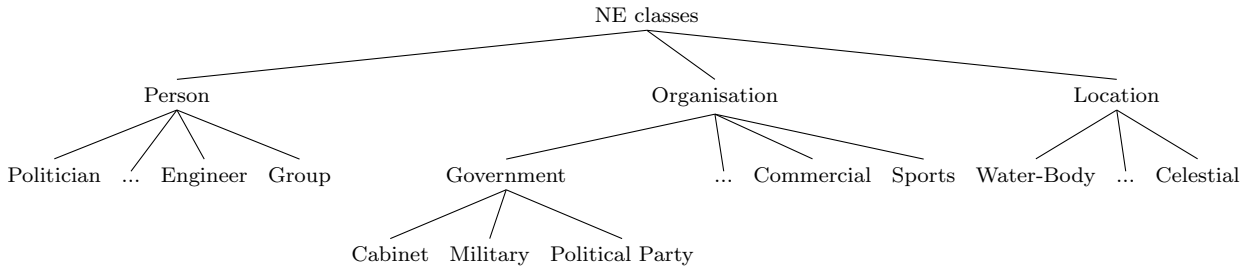

Figure 1.2: An example of fine-grained multi-levels tagset

Fine-grained NER plays an obvious role in information retrieval, in order to obtain proper snippet data from search engines for instance. Guo et al. (2009) conducted a manual analysis on 1000 unique queries selected randomly from the search log of a commercial web search engine[1] and found that about 71% of user queries are about NEs. Therefore, understanding users' queries would have a positive effect on overall performance. Paşca (2007); Alasiry et al. (2012);

[1]Live search engine (now called Bing) was used

Eiselt and Figueroa (2013); Alasiry et al. (2014), are among those who have investigated the effective detection of NEs within queries.

The identification of the NEs, in Question Answering (QA), facilitates the fetching of important information and the introduction of facts about those questions (Lee et al., 2007). Looking closely at the types of questions, factoid-type questions become salient. For instance, questions such as 'Who is the CEO of Microsoft?' and 'Where is Manningtree?' ask for named entities. Therefore, having the ability to classify the Microsoft CEO, i.e. Satya Nadella, and Manningtree as 'person.business' and 'location.town' respectively is semantically deeper and more helpful to QA systems than tagging with 'person' and 'location' respectively. In addition, among 200 factoid questions prepared for Text Retrieval Conference TREC-8, 80% of questions were asking about NEs (Voorhees and Tice, 1999). Moreover, Noguera et al. (2005) comprehensively studied the effect of the NER in a QA system to reduce the amount of data fetched. They found that, reliance on a NER component in QA reduces the retrieved data by 60% without significant data loss. Several studies use NER in QA, such as Mollá et al. (2006); McNamee et al. (2008); Mendes et al. (2010); Lee et al. (2006). Ferrández et al. (2007) showed the majority of questions included in the Conference and Labs for the Evaluation Forums (CLEF) 2004 and 2005 have one or more NEs. Therefore, the accurate fine-grained NER directly affects the accuracy of QA as virtually every contemporary QA system would fail to perform well in the absence of NER (Mollá et al., 2006).

Fine-grained NER is not only important for the tasks mentioned, but it can be used instead of other NLP tasks to deliver similar benefits; e.g. Text Summarisation (Nobata et al., 2002), Topic Detection (Ng et al., 2007) and Machine Translation (Babych and Hartley, 2003).

Therefore the motivation of this research is to broaden both the capacity and domain of Arabic NER, going beyond the newswire domain by undertaking the problem of fine-grained NER in four tuples: first, deciding upon the fine-grained taxonomy to be used; second, developing the necessary resources, i.e. a fine-grained annotated named entity corpora and gazetteer, from diverse resources; third, building a reliable fine-grained NER by relying on supervised ML techniques is major goal of this research. Therefore, several issues should be investigated via controlled experiments. These include: the annotation schemes, the probabilistic models and set of features to be involved. The classification process of fine-grained classes becomes more

difficult, because capturing informative evidence from local contexts is a non-trivial task. For example: consider the following two sentences:

1. (لعب كريستيانو رونالدو دورا مهما في بلوغ البرتغال مونديال /lʕb krystyAnw rwnAldw dwrA mhmA fy blwγ AlbrtγAlmwndyAl 2014/ 'Cristiano Ronaldo played an important role in achieving Portugal the 2014 World Cup')

2. (لعب ستيف جوبز دورا مهما في إنعاش شركة أبل /lʕb styf jwbz dwrA mhmA fy InʕAš šrkħ Obl/ 'Steve Jobs played an important role in the recovery of Apple')

Although, 'Cristiano Ronaldo' and 'Steve Jobs' are tagged in different fine-grained classes (athlete and businessman respectively), they appear in almost similar context. Therefore, (as the fourth tuple) this research investigates a new set of features, by moving beyond the local context and the sentence boundary.

1.2 Research Questions and Hypothesis

The difficulties surrounding the development of fine-grained NER for Arabic, as mentioned in the previous section, motivate the work presented in this thesis. Since the thesis will focus on fine-grained NER for Arabic, there are several research questions the researcher will strive to answer, as follows:

(RQ1): How can annotated fine-grained NE resources, such as corpora and gazetteer be created, to enable supervised fine-grained NER?

a) What is the source of knowledge?

b) How should the fine-grained NE semantic classes be established?

c) How should the annotation process be undertaken?

d) What is the most suitable annotation scheme for annotating NE phrases?

(RQ2): Which machine learning method is the most efficient in implementing fine-grained NER system?

(RQ3): How can informative features that go beyond the local context be defined and extracted, whilst also capturing the semantic differences between fine-grained classes?

(RQ4): How can global evidences that go beyond the sentence boundary be captured, in order to enhance the performance of fine-grained NER?

The research questions above delimit the scope of the research work and describe the primary aim of building a fine-grained NER for Arabic textual data. The work commences with an important research stage, with the aim of developing the required resources to enable supervised learning. The first question emphasises the need to first select information from an appropriate source of knowledge; whilst in parallel specifying the granularity of fine-grained semantic classes as directed towards a building tagset. This stage also involves selecting a suitable encoding scheme to represent the annotated entities. Since development of the data set is expected to involve proposing a novel solution, evaluating the efficiency of the data set developed is important.

The second research question requires the researcher to investigate efficient machine learning technology that is pertinent to the task. Since the number of semantic NE classes should be much greater than in a traditional system, in-depth and careful selection of the technology element is crucial.

The third question relates to features engineering. To answer this question, an investigation extracting informative features which have the ability to capture the semantic differences between fine-grained classes will be conducted.

The fourth question focuses on the investigation of the way of capturing global evidence with no restriction to the sentence boundary by exploiting the richness of the Arabic raw textual data.

The research questions mentioned above have raised the following hypotheses, which form the road map for the research:

Hypothesis 1: Online resources such as Arabic Wikipedia, which is a relatively open-domain collaborative encyclopaedia, can be exploited to develop an annotated scalable fine-grained NE gazetteer and corpus automatically. **(Related to RQ1)**

Hypothesis 2: Supervised machine learning techniques, utilising a sufficient amount of training data, can be employed to build a fine-grained NER system for Arabic textual data. **(Related to RQ2)**

Hypothesis 3: Instead of relying on language-independent window-based features representation (i.e. n-gram features representation), utilising the language-dependent approach by relying on the dependency-based representation of the features is a promising approach that goes beyond the size of window-based representations, especially for long Arabic sentences. **(Related to RQ3)**

Hypothesis 4: Utilising Arabic raw textual data to extract global features, by performing token-level hierarchy-based clustering allows the capture of global evidences beyond the sentence level that will boost the fine-grained NER performance. **(Related to RQ4)**

1.3 Contributions

The contributions of this thesis are presented as follows:

1. Investigating and studying the nature of Arabic NEs by exploring and defining their density, length, types, structures and semantic distribution, and then conducting a corpus-based evaluation in order to study the characteristics and properties of NEs across corpora.

2. Developing a methodology that exploits the richness of Arabic Wikipedia in order to automatically create a scalable fine-grained corpus and gazetteer. This resulted in:

   (a) WikiFANE$_{Auto}$: a fine-grained corpus of size 2M tokens

   (b) WikiFANE$_{Gazet}$: a fine-grained gazetteer comprises of 68355 entities

   In addition, two manually-created gold-standard fine-grained corpora from different genres were developed and this resulted in:

   (a) NewsFANE$_{Gold}$: a newswire-based fine-grained corpus of size 170K tokens

   (b) WikiFANE$_{Gold}$: a Wikipedia-based fine-grained corpus of size 500K tokens

3. Developing and evaluating a fine-grained NER for Arabic by learning two different supervised machine learning algorithms (i.e. Maximum Entropy (ME) and Conditional Ran-

dom Fields (CRF)) and investigating the effects of design decisions including: encoding schemes; and injecting external knowledge (i.e. gazetteer).

4. Presenting the development and the evaluation of a novel approach to representing the features by relying on the dependency structure, this involves:

   (a) Identifying the limitations of the current window-based representation;

   (b) Utilising the dependency structure of the sentence, working toward achieving the dependency-based representation of the features.

5. Exploiting the unstructured textual data with the intention of developing and evaluating a hybrid-based approach to fine-grained NER by performing word-level text clustering relying on Brown's (1992) hierarchal representation of clusters.

1.4 Publications based on the Thesis

The substantial ideas of this thesis have peer-reviewed and published in the following publications in chronological order:

- F. Alotaibi and M. Lee, "A Hybrid Approach to Features Representation for Fine-grained Arabic Named Entity Recognition", In Proceedings of the 25th International Conference on Computational Linguistics (COLING), p984-995. Dublin, Ireland, August 23-29, 2014.

- F. Alotaibi and M. Lee, "Automatically Developing a Fine-grained Arabic Named Entity Corpus and Gazetteer by utilizing Wikipedia", In Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP), p392-400. Nagoya, Japan, October, 2013. (acceptance rate: 23.4%)

- F. Alotaibi and M. Lee, "Towards an Automatic Development of Named Entities Corpus from Arabic Wikipedia", In Second Workshop on Arabic Corpus Linguistics: Posters, p60-61, Lancaster University, UK, July 22nd. 2013.

- F. Alotaibi and M. Lee, "Detecting Named Entities in the Arabic Wikipedia" Linguistica Communicatio: International journal of Arabic language engineering & General Linguistics, 5:109-126. 2013

- F. Alotaibi and M. Lee, "Mapping Arabic Wikipedia into the Named Entities Taxonomy", In Proceedings of the 24th International Conference on Computational Linguistics (COLING): Poster, p43-52. IIT, Mumbai, India, December 8-15. 2012. (acceptance rate: 27%)

- F. Alotaibi and M. Lee, "Using Wikipedia as a Resource for Arabic Named Entity Recognition", In the 4th International Conference on Arabic Language Processing (CITALA12), Rabat, Morocco, May 2-3. 2012. (acceptance rate: 27.7%)

1.5 Thesis Structure

The reminder of this thesis is divided into four major parts comprised of eight chapters (excluding the introduction chapter); the structure of these chapters is as follows:

Part II: BACKGROUND

Chapter 2 forms the first portion of the background segment, where it presents the background to the Arabic language from a linguistic point of view. It shows the nature of Arabic, including the scripting system, morphology and syntax characteristics. Moreover, challenges associated with Arabic are presented. These challenges concern NER tasks complicated by the absence of capitalisation and short vowels, data sparseness, transliteration problems, and ambiguity. This chapter concludes by presenting the possible types and different structures of Arabic NE phrases, to facilitate understanding of their nature.

The second portion of the background section is presented in Chapter 3, where a comprehensive literature review is presented as background to the Arabic NER. The review covers several issues, including the overview of the task of NER, and the available resources (i.e. corpora, lexical resources and tools). The chapter ends with a chronological review of the approaches used to develop Arabic NER with: handcrafted rules based, machine learning based, or hybrid based approaches.

Part III: FINE-GRAINED RESOURCE CREATION (Related to Hypothesis 1)

Chapter 4 is dedicated to presenting the approach advised to develop a scalable fine-grained NE gazetteer comprising 68355 entities. Arabic Wikipedia is selected as the source of knowledge from which to develop the desired lexicon. By formulating this task as a document classification problem, several issues are addressed in this chapter. These issues begin with those associated with defining the semantic tagset, and are followed by consideration of the strategy advised to annotate predefined set documents to be used as training data. The representation and engineering of features for document classification are also covered.

In Chapter 5, the methodology advising the creation of a publically-open fine-grained NE corpus is discussed. The first approach to developing these resources involves utilising the textual data of Arabic Wikipedia to develop a scalable corpus automatically (more than 2M of tokens). However, a decision was made to also develop two smaller corpora manually from two genres, i.e. newswire and Wikipedia. The reason for this development is to have the ability to, firstly, study the nature of NE phrases in different genres and with different annotation methods, i.e. automatically and manually; and secondly, to conduct a comprehensive evaluation, by analysing the behaviour of the different probabilistic models and sets of features for each corpus with comparable evaluation. Moreover, this chapter presents corpus-based evaluation across corpora for different metrics, including: density, length, phrase structures and semantic class distribution of NER phrases.

Part IV: FINE-GRAINED NAMED ENTITY RECOGNITION (Related to Hypothesis 2, 3 and 4)

Chapter 6 presents, in detail, the development of a pipeline structure of fine-grained NER. Since there are no comparable results for fine-grained NER, a baseline model based on Maximum Entropy (ME) is established. Investigation of the performance of the NER system by learning from a different statistical model, i.e. Conditional Random Field (CRF), is undertaken. Several design decisions, including applying external knowledge and the encoding scheme are presented. This chapter, and the following two chapters conclude with a comprehensive error analysis, including a confusion matrix, error based on the length of the NE phrase, and error within fine-grained classes of the same parent.

The development of fine-grained NER presented in Chapter 6 relies on window-based representation of the features (n-gram representation). Meaning that, to make a decision to classify a token at position (i) as NE or not, a window of five tokens (for instance) consisting of two tokens before and after the current one, including the token at position (i), is involved in the classification process. Since, this research undertakes the problem of fine-grained Arabic NER for a large number of semantic classes (i.e. 50 classes), the window-based features representation represents a limitation when trying to capture informative semantic clues, especially with long sentences (taking into consideration that the average length of Arabic sentence varies from 31 to 38 tokens as will be seen in Chapter 5). Therefore, in Chapter 7, instead, dependency-based features representation is investigated and applied to capture features that go beyond the size of the window-based representation. A hybrid approach, exploiting both window-based and dependency based feature representation is also a promising option. This chapter concludes in a similar way to Chapter 6 by analysing the errors present.

Chapter 8 presents further investigation to exploit global evidences that go beyond the sentence boundary. Therefore, collections of unannotated large textual data (i.e. raw text) are exploited by performing word-level hierarchy clustering. The assumption is that, similar words appear in similar contexts. Applying such features at the top of those presented in Chapter 7 yields improvement in the overall performance. Error analysis is also discussed in detail at the end of this chapter.

Part V: CONCLUSION

Chapter 9 concludes the thesis by elaborating on how the approaches in previous chapters have satisfied the overall goals of the research and delivered contributions to the field of knowledge. This chapter also presents potential future work and possible research directions for Arabic NLP in direct relation to NER.

Part II

BACKGROUND

The background part is divided into two chapters. Chapter 2 presents the background of Arabic as a target language of this research. Chapter 3 extensively reviews the literature in relation to the Arabic NER task.

Chapter 2

Background of the Target Language

Chapter Synopsis

This chapter focuses on reviewing certain characteristics of Arabic in relation to NER. For comprehensive linguistic details, Habash (2010) provides an introduction to Arabic Natural Language Processing, and recently Darwish and Magdy (2014) focus on different issues related to Arabic Information Retrieval. In Section 2.1, characteristics of the Arabic language are examined from a linguistic viewpoint. This is followed, in Section 2.2, with a description of the linguistic challenges presented by Arabic named entities. In Section 2.3, different types and structures of Arabic NEs are discussed. This chapter as a whole aims to provide background knowledge about the language targeted by this research.

2.1 Characteristics of the Arabic Language

The Arabic language has developed over the centuries from its original classical form, into what is now described as Modern Standard Arabic (MSA). MSA is the official language of the Arab world and in its written form is based on Classical Arabic (CA) in its syntax and morphology. However, region-specific colloquial spoken Arabic can differ widely in nature from one place to another. Although CA and MSA share many common features, MSA tends to have a more modern vocabulary and even loanwords (Ryding, 2005). Moreover, short vowels do not appear

in writing; where these are omitted the reader should use the context to identify the correct pronunciation (Shaalan, 2010).

2.1.1 Scripting Nature of the Language

Arabic has 28 different isolated letters which are connected form right to left. Each isolated letter has a different shape according to its position in the word, i.e. initial (بـ), middle (ـبـ), last (ـب) or separated (ب). There are eight other letters (or letter forms) which are (إ /I/), (أ /O/), (آ /$\widehat{A}$/), (ء /'/), (ؤ /W/), (ئ /$\widehat{y}$/), (ة /$\hbar$/) and (ى /$\acute{y}$/). Fifteen of the letters contain dots to differentiate them from other letters (Habash, 2010). Letters may or may not have diacritics to represent the short vocal sounds of the corresponding vowel (i.e. (ـَ/a/), (ـِ /i/) and (ـُ /u/)) written above or below a letter. Special forms (i.e. ligatures) for some character sequences and kashida, which is a symbol that extend the length of words, are often employed in printed text. Figure 2.1 demonstrates some of these orthographic features.

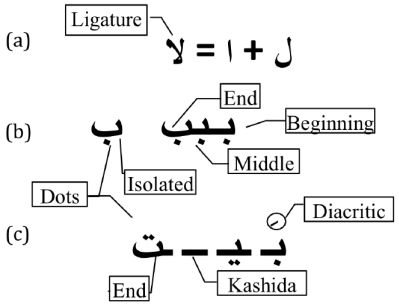

Figure 2.1: (a) Example of a ligature, (b) the different shapes of the letter (ب /b/) and (c) example of a diacritic, kashida, and three letters which are distinguishable from each other only by dots (Darwish and Magdy, 2014)

2.1.2 Arabic Morphology

Arabic is a Semitic language which exhibits nonconcatenative morphology (McCarthy, 1981), which is highly systematic (Ryding, 2005). A word is a composition of root and pattern. The former and the latter are representing the consonants and the pattern of vowels respectively. A quick overview of the root, pattern, derivational and inflective properties is given below.

2.1.2.1 Root and pattern

The root and pattern are key concepts in Arabic morphology. They interlock to form the final shape of the word. Roots, as discontinuous morphemes, consist mainly of three or four radicals, i.e. consonants, and, rarely, five (Farghaly and Shaalan, 2009; Ryding, 2005; Darwish, 2002). The root and pattern possess the lexical and grammatical meanings respectively. Beesley (1996) shows that Arabic has almost 5000 roots, while Ryding (2005) estimates there to be between 5000 and 6500. Other researchers have stated that the combined number of noun and verb roots could be as many as 10000 (Darwish, 2002). In the case of patterns, there are estimated to be about 400 different patterns (Beesley, 1996).

2.1.2.2 Derivation

This is the process of word formation from the root. The actual derivation process occurs when combining a specific consonantal root with the desired pattern. Moving vowels between consonants results in the creation of differently derived words (Ryding, 2005).

2.1.2.3 Inflection

The inflection of Arabic words is mainly influenced by the position of a given word in its particular context (Ryding, 2005). Different inflectional categories, i.e. features, apply to nouns, verbs and pronouns. Four inflectional features are applied to nouns and adjectives: gender, number, case and definiteness. Verbs have a larger number of features: aspect, person, voice, mood, gender and number. Finally, pronouns usually possess four different features: person, gender, number and case (Ryding, 2005).

2.1.3 Arabic Syntax

Arabic has two types of sentence: nominal and verbal, depending on which Part of Speech (POS) forms the first word in the sentence. A nominal sentence, which is referred to as an equational sentence, contains no verb; instead it is formed of a subject and a predicate. It varies from very simple forms, which consist only of noun and adjective, to more complicated ones, in which the subject is a compound of two words and the predicate is another equational sentence (Ryding, 2005). By contrast, the verbal sentence starts with a verb, and follows different structures and

orders. The simplest form is a verb + subject pronoun, where the subject pronoun is part of the verb as a result of the inflectional process, e.g. (لعبت - /libt/ - 'She played') or even pro-drop one, e.g. (لعب - /lib/ - 'He played'). The standard form of the verbal sentence follows a verb-subject-object (VSO) structure (Ryding, 2005). This is also applicable if the sentence does not have a direct object, in the case of verb intransitivity. Verbal sentences have a variety of word orders, and Arabic is considered a relatively free word ordering language (Farghaly and Shaalan, 2009; Ryding, 2005; Shaalan, 2005). Subject-verb-object (SVO) is another variation of the verbal structure, where the verb follows the subject. This form is frequently used in the headlines of Arabic newspapers (Ryding, 2005). Alternatively, the object could follow the verb to form one of the variant shapes which is verb-object-subject (VOS). Other variations, such as OSV and OVS, are less often used in MSA (Diab et al., 2008) but they are linguistically valid.

2.2 Challenges Concerning Arabic Named Entities

There are several challenges to the linguist relating to Arabic NEs; these are described below.

2.2.1 Absence of Capitalisation

In contrast to English, Arabic letters are written in a cursive way with no capitalisation. This can cause some difficulty when dealing with, and identifying, proper nouns in Arabic. In Arabic there is no internal lexical sign to distinguish whether a word is a noun, a verb, a preposition or even an adjective; instead, this must be determined by examining the word in context.

2.2.2 Absence of Short Vowels

MSA is written literally, without considering the addition of diacritics; this has a major impact on the approach of the morphological analyser, who tends to return all possible morphological variations for a given word due to the lack of diacritics. Unfortunately, MSA text discards these

as a matter of course, thus making it necessary for the reader to refer to the context in order to predict the correct vocalisation and pronunciation (Ryding, 2005).

2.2.3 Data Sparseness

The nonconcatenative morphology of Arabic results in increasing the data sparseness (McCarthy, 1981). On the first count, NEs of the type 'proper nouns' are commonly agglutinated by prefixes, and are ordered as follows:

[Question + [Conjunction + [Particle + [Definite article + [Proper Noun]]]]]

Conversely, NEs formed by complex phrases, e.g. common nouns, might have both prefixes and suffixes. The general representation of the order of the affixes for each token of the NE is expressed as follows:

[Question + [Conjunction + [Particle + [Definite article + [Base] + Proclitic]]]]

Detecting NEs over the full form of tokens can lead to data sparseness. Two possible ways to resolve this issue are either by stemming or tokenizing the token. Although the former method is apparently faster, it leads to a loss of valuable information from the affixes. The latter approach solves this problem, by separating the suffixes with spaces.

2.2.4 Transliteration Problem

Non-Arabic NEs of the type 'proper nouns' have various ways of being mapped into the Arabic alphabetical system. The absence of a strict mapping strategy raises a problem when applied to a given proper noun, as there are a number of accepted forms of Arabic equivalence. This directly affects the task of Arabic NER (Farghaly and Shaalan, 2009). For example, a proper noun such as 'Birmingham' could be expressed using three acceptable forms such (برمنقهام /brmnqhAm/) , (برمنغهام /brmnϒhAm/) and (برمنجهام /brmnjhAm/).

2.2.5 Ambiguity

Not all NEs are of which simply analysed as 'proper noun'. Instead, there are great levels of ambiguity, because the NE phrases can be freely formed by different POS such as 'common nouns', 'adjectives' or more complex phrases of more than one token. The following shows some linguistic cases which also contribute to increase the level of ambiguity.

2.2.5.1 Diminutives

These result from a change to the internal structure of the noun. Some of the usages of diminutives involve scaling down the size and degradation. A number of diminutive nouns are used as personal names such as (عبيد / ʕbyd/ 'Obaied'), (صويلح /SwylH/ 'Sowaileh'), (دريهم /dryhm/ 'Doraihim') and so on.

2.2.5.2 Participles

One of the nominal forms of Arabic verbs is the participle. Participles are either active or passive and can be used as NEs. Both variant participles can be used as personal NEs. For example (as seen in Table 2.1):

Table 2.1: An example of the participles ambiguity in Arabic NE

Participle type	Personal NE	Verb
Active	(حامد /HAmd/ 'Hamid')	(حمد /Hmd/ 'Hamad')
Passive	(محمود /mHmwd/ 'Mahmood')	(حمد /Hmd/ 'Hamad')

2.2.5.3 Inflected verbs

Some inflected verbs have shared functions as both verbs and personal NEs. For example: (يزيد /yzyd/ 'Yazeed' (proper noun) or 'increases' (inflected verb)) and (أحمد /OHmd/ 'Ahmad' (proper noun) or 'I thank' (inflected verb)).

2.3 Arabic NE Types and Structures

This section reviews both the type and structure of Arabic NEs from the language point of view.

2.3.1 Types of Arabic Named Entities

Arabic NEs can be classified into personal, non-personal and borrowed NEs. This section provides a brief description of the types of Arabic NEs.

2.3.1.1 Personal Named Entities

Personal NEs can be classified into five main types: given names, epithetons, teknonyms, patronymics and relative adjectives.

- **Given names:**

  A given name is also called an (اسم /Asm/ 'Given name') and is the name given at birth. Given names usually have an inherent meaning, such as (عبد الله / ʕbd Allh/ 'Abdullah - the slave of god'). Other names have been transformed from other Semitic languages, such as (إبراهيم /IbrAhym/ 'Ibrahim').

- **Relative adjectives:**

  Sometimes called (نسبة /nsbħ/ 'Relative adjective'), these are derived from names associated with a profession, a religion, a particular geographical place or a tribal affiliation. They are formed by adding the letter (ي /y/) to the end of the noun or verbal noun, i.e. the infinitive. The resulting relative adjective relates semantically to its origin. For instance, the relative adjective (مكي /mky/ 'related to Mecca') is related to the noun (مكة /mkħ / 'Mecca'), which is a place name.

- **Epithetons:**

  These are considered nicknames and referred to as (لقب /lqb/ 'Epithetons'). They can replace a given name in an appropriate context. For instance, (الفاروق /AlfArwq/ 'Dis-

tinguish between truth and falseness') is an epitheton of the second caliph (عمر بن الخطاب /ςmr bn AlxTAb/ 'Umar Ibn Al Khattab').

- **Teknonym names:**

  These are names that have been derived from the child's given names. They consist of a particular word, i.e. (أبو /Obw/ 'father of') and (أم /Om/ 'mother of'), followed by the given name of the first child.

- **Patronymic names:**

  Unlike teknonym names, these are derived solely from the father's name and follow a general rule like 'son of A' for a male and 'daughter of A' for a female, where 'A' is the father's given name. For example (ابن تيمية /Abn tymyħ/ 'Ibn Taimiyah').

2.3.1.2 Non-Personal Named Entities

These include, but are not limited to, the names of particular places, such as countries, cities, geographical features etc. Some require the definite article whilst others have no definite prefix (which is called diptote). For example (مصر /mSr/ 'Egypt'), (الدار البيضاء /AldAr AlbyDA'/ 'Casablanca') and (الربع الخالي /AlrbςAlxAly/ 'Empty Quarter').

2.3.1.3 Borrowed Names and Acronyms

English proper names and acronyms are transliterated into Arabic as proper nouns. There are many such names and acronyms that are used for persons, organisations and locations. For example (مارك /mArk/ 'Mark'), (يونيسكو /ywnyskw/ 'UNESCO') and (بيرمنقهام /byrmnqhAm/ 'Birmingham').

2.3.2 Different Structures of Arabic Named Entities

Arabic NEs are formed from a diverse range of elements, covering simple phrases and more complex ones as described below.

2.3.2.1 Simple Phrases

This category involves NEs in which all tokens fall into the category of proper nouns (NNP) such as (محمد /mHmd/ 'Mohammed').

2.3.2.2 Complex Phrases

In Arabic only a percentage of NEs can be identified as simple proper nouns. For instance, there is a wide range of organisational entities whose names are comprised of more complex phrases. These phrasal names have different nominal structures. Represented below is a summary of the most important phrases used for NEs:

- **Noun Phrase (NP): [NN or NNS]**

  Although a token of the type singular common noun (NN) or plural/dual common noun (NNS) seem to be simple, there is also often ambiguity as to whether it should be considered as a common noun or a NE: for example: (وردة /wrdħ / 'Flower - or a female name Wardah') and (رمضان /rmDAn/ 'an Islamic month - or a male name Ramadhan').

- **Noun Phrase (NP): [NN + JJ]**

  The adjective is used in NEs following the head noun and concurs with the noun in both definiteness and case such as in the phrase (البيت الأبيض /Albyt AlObyD/ 'the White House').

- **Noun Phrase (NP): [NN + NN]**

  One of the most common phrases consists of compound nouns in the syntactical form of construction and is called (إضافة /IDAfħ/ 'Idafa'). The head noun is called a possessor whilst the second noun is called the possessed. The possessor and the possessed have

construct and genitive case respectively. An example of this structure is (مجلس الأمن /mjls AlOmn/ 'Security Council').

- **Noun Phrase (NP): [Idafa chain (NN + NN + ... + NN)]**

  The two nouns of the Idafa construction can be preceded by new head nouns in a recursive manner. The result is referred to as the 'Idafa chain': for example (هيئة كبار العلماء /hyħ kbAr AlʕlmA'/ 'The Council of Senior Scholars').

- **Noun Phrase (NP): [Idafa + JJ]**

  Adjectives could be attached to the end of the Idafa and concur with the head of the Idafa in case; whilst corresponding with the possessed in the definiteness. This type of noun phrase is typically used for organisational and governmental NEs: for instance, (منظمة العفو الدولية /mnǍmħ Alʕfw Aldwlyħ / 'Amnesty International').

- **Noun Phrase (NP): [NP + CC + NP]**

  Using a conjunction (CC) with a NE makes it difficult to distinguish whether it is part of the entity or two separate entities. Thus organisations and governmental departments may potentially contain a conjunction in their names: for example, (جمعية البر والإحسان الخيرية /jmʕyħ Albr wAlIHsAn Alxyryħ / 'Welfare and Charity Foundation')[1].

- **Noun Phrase (NP): [NP + PP[2]]**

  Prepositions (IN) such as (ل /l/ 'for') are commonly used to name organisations and are also considered to be part of the organisational name: for example (أقوات للصناعات الغذائية /OqwAt llSnAʕAt AlϒðAyħ / 'Aquat for Food Industries').

- **Further complex phrases:**

  NEs in Arabic can also be long and form more complex phrases. Thus conjunctions and

[1]The use of the conjunction in NE phrase complicates the task. There are two interpretations whether the conjunction is considered as a separator for two entities (e.g. Microsoft and Yahoo) or to join them to represent single NE (e.g. Marks and Spencer).

[2]PP stands for Prepositional Phrases

prepositions are used with nouns such as: (مؤسسة الملك عبد العزيز ورجاله للموهبة والإبداع /mWssħ Almlk ʕbd Alʕzyz wrjAlh llmwhbħ wAlIbdAʕ / 'the foundation of King Abdul Aziz and his men of talent and creativity').

The sheer variety of NE phrases demonstrates that they cannot be analysed simply as proper nouns. Consequently, the detection of such phrases becomes challenging, especially when taking into account the absence of important orthographical signs such as capitalisation.

2.4 Chapter Summary

This chapter forms the first part of the literature review and has dealt with the language background of Arabic NEs, detailing language type, structure and the challenges involved in identifying these entities accurately. In the next chapter, we will present the second part of the literature review where the state-of-the-art techniques of NER are presented.

Chapter 3

Background of Arabic NER

Chapter Synopsis

In the previous chapter, the background of Arabic as a target language of this research was presented. In this chapter, a detailed literature review of the Arabic NER will be undertaken. The aim of the NER system is to detect and classify NEs into semantic classes, and therefore this chapter will present several issues. Section 3.1 provides an overview of the NER task which reviews some aspects including the definition of the NE, the semantic tagset used in the literature, the formal definition and the evaluation methods for NER systems. In Section 3.2, there will be a presentation of available resources (including corpora, lexical resources, the environment for developing NER, and essential pre-processing tool for Arabic). In Section 3.4, there will be a discussion of different approaches to addressing Arabic NER.

3.1 An Overview of NER

3.1.1 What is the NE?

Although early in 1991, Rau (1991) proposed a method of extracting company names, the actual term Named Entity (NE) was only coined and introduced later at the Sixth Message Understanding Conference (MUC-6) in 1996, to refer to "unique identifiers of entities". In 2000, Petasis et al. (2000) limited the scope of NE to "a proper noun, serving as a name for something or someone". This is similar to the definition given by Jurafsky and Martin (2000), which was "anything that can be referred to with a proper name". Alfonseca and Manandhar (2002)

defined NE as "the task of classifying unknown objects in known hierarchies that are of interest to us being useful to solve a particular problem". In 2002, the Conference on Computational Natural Language Learning (CoNLL) (Tjong Kim Sang and De Meulder, 2003) introduced the NER shared task, presenting the definition of NE as "phrases that contain the names of persons, organisations and locations". Automatic Content Extraction (ACE) differentiates between NE as "an object or set of objects in the world" and the mention as "a reference to an entity" (ACE, 2003). Nadeau and Sekine (2007) argued that NE is restricted to those entities that are referred to as rigid designators, following Kripke (1972) definition of rigidity as, "a designator d of an object x is rigid if it designates x with respect to all possible worlds where x exists, and never designates an object other than x with respect to any possible world".

We can observe, from this variation, that there has been some difficulty and widespread disagreement in agreeing upon a single and clearly denoted definition of NE. Therefore, in this thesis, and to avoid ambiguity, we apply the definition of NE presented by Jurafsky and Martin (2000), which is "anything that can be referred to with a proper name" regardless of whether this name is simply analysed as a proper noun such as (لندن /lndn/ 'London') or presented in complex structure such as (مجلس القضاء الأعلى /mjls AlqDA' AlOcl*ý*/ 'The Supreme Judicial Council'); the key determiner being that it refers to particular object that exists.

3.1.2 The Semantic Tagset of NER

The task of NER focuses on delimiting the boundary of the NEs and assigning them an appropriate semantic class (Grishman and Sundheim, 1996; Chinchor and Robinson, 1997). There are a number of different tagsets in the literature which have been devised and widely used. These are classified into coarse-grained (i.e. single level of small predefined classes) and fine-grained (i.e. hierarchal representation of the classes with at least two levels) tagsets, as discussed in detail below.

3.1.2.1 Coarse-grained Tagsets

In the literature, three traditional tagsets have been used extensively. MUC-6 (Grishman and Sundheim, 1996) was the first event in which the role of NER was established. MUC-6 defines three elements of NEs: (1) ENAMEX (which includes personal, locational and organisational names); (2) NUMEX (i.e. numerical expressions); (3) TIMEX (which tags periods, times and dates). Examples of studies using this tagset for Arabic NER are as follows: (Elsebai, 2009; Benajiba et al., 2009b; Abdul-Hamid and Darwish, 2010; Asharef et al., 2012).

In 2002, CoNLL introduced the first shared task on language-independent NER (Tjong Kim Sang and De Meulder, 2003). The tagset employed during this conference were three basic types of NE, i.e. person (PER), location (LOC) and organisation (ORG). Any NEs that do not belong to one of those tags will be tagged as 'miscellaneous' (MISC). This tagset has been used frequently in Arabic NER, including by (Benajiba and Rosso, 2007; Koulali and Meziane, 2012; Mohammed and Omar, 2012; Morsi and Rafea, 2013).

The ACE program commenced in 2003 as series of events aiming to stimulate and benchmark research in information extraction. Its scope is broader than both MUC and CoNLL, with the focus being wider than simply detecting and classifying NEs, covering relations between NEs as well as the event detection. In relation to NER, ACE provided a tagset which differs slightly from CoNLL. Five coarse-grained categories are defined to tag entities, e.g. person (PER); organisation (ORG); location (LOC); facility (FAC); and geographic and political (GPE). One year later, in 2004, two new additional types were added to tag vehicles (VEH) and weapons (WEA). Moreover, ACE provided two levels of the type taxonomy, i.e. coarse- and fine-grained. For example, an organisation classed as coarse-grained is further classified into sub-types: Government; Non-Governmental; Commercial; Educational; Media; Religious; Sports; Medical; Science; Entertainment. In ACE (2004) and ACE (2005), there is a total of 45 subclasses. Several studies including (Benajiba et al., 2008b,a; Benajiba and Zitouni, 2009; Benajiba et al., 2010) have merely used the coarse-grained level of this tagset.

The Approximate Overlapping of Different Coarse-grained Tagsets: At the coarse-grained level, the three widely-used tagsets have overlapped; these represent three important types, i.e. PER, ORG and LOC (see Table 3.1). Both CoNLL and ACE do not show

any interest in tagging numerical expressions, i.e. time, date, currency and percentage, whereas MUC does. Therefore, researchers who work at the coarse-grained level and perform evaluations across corpora typically focus on the three agreed upon types.

Table 3.1: The approximate overlapping of different coarse-grained tagsets

MUC	CoNLL	ACE
PER	PER	PER
ORG	ORG	ORG
LOC	LOC	LOC & GPE & FAC
-	MISC	-
Periods & Time & date	-	-
Currency & percentage	-	-
-	-	WEA & VEH

3.1.2.2 Fine-grained Tagsets

Current approaches to Arabic NER have provided, and been applied to, a limited number of semantic classes. Although the semantic classes of these tagsets are considered to be coarse-grained, a number of studies have been used a subset of these coarse classes, including (Abdul-Hamid and Darwish, 2010), as focusing on PER, ORG and LOC classes.

When it comes to English (and other languages), the semantic classes have been extended into two dimensions, as discussed below (The actual classes for each tagset are presented in details in Table 3.2):

Single-class Extension Fleischman (2001) argues that the 'IdentiFinde' tool, which recognises the coarse-grain classes proposed by Bikel et al. (1999), will prove more helpful once a greater number of finer classes have emerged, even though his system has resulted in high performance. Due to its simplicity in comparison with a personal class, Fleischman (2001) proposed an extension of the 'location' class into eight subclasses.

A further extension to the previous work has been proposed by Fleischman and Hovy (2002), in which the personal class was decomposed into a number of finer classes. Eight sub categories were defined, based on their high frequency in the corpus, as well as in regard to their usefulness in applications (e.g. Question Answering). More recently, Giuliano and Gliozz (2008) have built wider sub categories using WordNet for the class 'PER'. The difference in their work is that the

subclasses are built in a hierarchical manner extracted from WordNet and the classes involved need to have been populated by at least 40 instances of proper nouns. This has led to the development of a total of 21 subclasses.

A similar approach has been taken by Ekbal et al. (2010), in which WordNet was utilised to build subcategories. Instead of searching within WordNet to find proper nouns that fit into different classes, sets of patterns are used to extract potential personal nouns from textual data. These are then mapped into WordNet categories. As a result, a depth of eight levels has been developed, in which a total of 213 personal subclasses have been formed.

Generic Extension Unlike previous studies, the efforts detailed in this section have sought to extend the traditional tagset of NEs into finer semantic classes. Sekine et al. (2002) have proposed a hierarchical NE taxonomy that is very fine, comprising of 150 subclasses. The methodology used to construct the semantic classes relies on analysing the NEs in a newswire corpus, in addition to analysing the answer type for a set of questions used in the Text REtrieval Conference TREC-QA task. The WordNet noun hierarchy is also used to further shape these classes. Two years after this initial study, Sekine and Nobat (2004) added an additional 50 classes and decomposed others (such as 'disease' and 'numeric expression'). Although the spectrum of classes is wide, the specific description and definition of each class strives to avoid overlap and ambiguity, and therefore is not easy to define. This taxonomy has been applied to both English and Japanese.

A number of NLP applications (such as QA) have designed their own NE tagset, based on criteria believed to contain the most usefulness. Harabagiu et al. (2003) have developed a NER component, in which one level consists of 20 defined fine-grained classes involving: (Quantity, Number, Date, Person, Country, Other locations, City, Organisation, Authored-work, Product, Continent, Province, Quote, University, Price, Science-name, Acronym, Address, Alphabet, URI).

Li and Roth (2006), understanding that factoid type questions are asking about NEs, have defined a fine-grained taxonomy to answer certain types of questions. Although, this two-layer taxonomy covers 50 fine-grained classes of different types, some types are unrelated to NEs (e.g. definition, description, manner and reason). Based on the same trend, Brunstein (2002)

presented two levels of taxonomy, in which 29 answer types are subdivided into a total of 105 subtypes. A number of researchers have adopted this taxonomy, employing it for NE tagset (Nothman et al., 2008).

Nothman et al. (2009) have created fine-grained tagset to be used over the Wikipedia-based textual data. The tagset proposed by Brunstein (2002) is partially used, and 60 fine-grained semantic classes have been employed. An extension of this work has been presented by Balasuriya et al. (2009), and constitutes a gold standard annotated NE corpus developed from English Wikipedia. Unlike Nothman et al. (2009), the annotation schema has been extended to cover 96 fine-grained classes, while mapping to the four coarse-grained CoNLL classes for evaluation and comparison (i.e. PER; LOC; ORG; MISC).

Table 3.2 presents the semantic classes of the different tagsets presented in literature in relation to the NER task (the numerical classes, as presented by Sekine and Nobat (2004) are omitted for the sake of the presentation). (The square brackets '[]' are used to show the fine-grained sub-classes when applicable, and a **bold font style** is used to identify their parents).

Table 3.2: Different fine-grained tagsets

Publication	**Type**	**Classes**
(Fleischman, 2001)	Single-class (Location only)	Country, City, Street, Territory, Region, Water, Mountain, and Artefact.
(Fleischman and Hovy, 2002)	Single-class (Person only)	Athlete, Politician, Clergy, Businessperson, Artist, Lawyer, Scientist, and Police
(Ekbal et al., 2010)	Single-class (Person only)	213 classes (the classes names are not mentioned in the paper)
(Nothman et al., 2008)	Generic extension (One-level fine-grained)	60 classes (the classes names are not mentioned in the paper)
(Harabagiu et al., 2003)	Generic extension (One-level fine-grained)	Quantity, Number, Date, Person, Country, Other locations, City, Organisation, Authored-work, Product, Continent, Province, Quote, University, Price, Science-name, Acronym, Address, Alphabet, URI,

Continued on next page

Table 3.2 – continued from previous page

Publication	Type	Classes
(Li and Roth, 2006)	Generic extension (Two-level fine-grained)	**Abbreviation**[abbreviation, expression], **Description**[definition, description, manner, reason], **Entity**[animal, body, colour, creative, currency, disease/medicine, event, food, instrument, lang, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word], **Human**[group, individual, title, description], **Location**[city, country, mountain, other, state], **Numeric**[code, count, date, distance, money, order, other, period, percent, speed, temp, vol.size, weight]
(Brunstein, 2002)	Generic extension (Two-level fine-grained)	**Person**, Person Descriptor, **NORP**[Nationality, Religion, Political, Other], **Facility**[Building, Bridge, Airport, Highway_Street, Attraction], Facility Descriptor, **Organisation**[Government, Corporation, Educational, Religious, Political, Museum, Hotel, Hospital, Other], Organisation Descriptor, **GPE**[Country, City, State/province, Other], GPE Descriptor, **Location**[River, Lake_Sea_Ocean, Border, Region, Latitude-Longitude, Continent, Other], **Product**[Weapon, Vehicle, Other], Product Descriptor, **Date**[Date, Duration, Age, Other, Unmarked], Time, Percent, Money, **Quantity**[distance, area, volume, Energy, Speed, Temperature, Acceleration, Weight, Other], Ordinal, Cardinal, **Events**[War, Hurricane, Other], Plant, Animal, **Substance**[Food, Drug, Nuclear, Chemical, Other], Disease, **Work of Art**[Book, Play, Song, Painting, Sculpture, Other], Law, Language, **Contact info**[Address, Email, Phone, URL], Game
(ACE, 2005)	Generic extension (Two-level fine-grained)	**Person**[Individual, Group, Indeterminate], **Organisation**[Government, Non-Governmental, Commercial, Educational, Media, Religious, Sports, Medical-Science, Entertainment], **Location**[Address, Boundary, Water-Body, Celestial, Land-Region-Natural, Region-General, Region-International], **GPE**[Continent, Nation, State-or-Province, County-or-District, Population-Center, GPE-Cluster, Special], **Facility**[Building-Grounds, Subarea-Facility, Path, Airport, Plant], **Vehicle**[Land, Air, Water, Subarea-Vehicle, Underspecified], **Weapon**[Blunt, Exploding, Sharp, Chemical, Biological, Shooting, Projectile, Nuclear, Underspecified]
(Giuliano and Gliozz, 2008)	Single-class (Hierarchy-based Person only)	Businessman, **Performer**[Actor, Musician], **Scientist**[Chemist, Mathematician, Physicist, Biologist, Social Scientist], **Communicator**[Representative, **Writer**[Poet, Dramatist]], Health Professional, **Creator**[Film maker, **Artist**[Painter, Musician]]

Continued on next page

Table 3.2 – continued from previous page

Publication	Type	Classes
(Balasuriya et al., 2009)	Generic extension (Hierarchy-based fine-grained)	**Personal**[Fictional, Other, Person, Religious], **Organisation**[Army, Band, Broadcaster, Charity, Tribe, Corporation, Educational, Environmental, Government, Health-facilities, Hotel, Museum, Other, Political, Religious, Sport], **Location**[**GPE**[Admin Region, Country, Other, Region, Religious, State-Province, Suburb, City], Border, Continent, Forest, Geological-region, Island, Other, River, Space, Water], **Facility** [Airport, Attraction, Bridge, Building, Farm, Library, Other, Road, Stadium, Station], **Product**[Electronics, Food-Drink, Franchise, Other, Protocol, Software, Vehicle, Weapon, Website], Artefact, Vessel, **Work of Art**[Album, Book, Film, Newspaper-Magazine-Journal, Other, Painting, Play, Sculpture, Song-Poem-Music, TV Show]], **Misc**[Award, Courtcase, Currency, **Event**[Concert, Natural Disaster, Other, Season, Sports, War-Battle], Game, Genre, Language, Law, Other, **Numeric**[Age, Cardinal, Date, Duration, Money, Ordinal, Other, Percent, Periodic, Quantity, Time, Unmarked]

Continued on next page

Table 3.2 – continued from previous page

Publication	Type	Classes
(Sekine and Nobat, 2004)	Generic extension (Hierarchy-based fine-grained)	**Name** [Name_Other, Person, God, **Organisation**[Organisation_Other, International_Organisation, Show_Organisation, Family, **Ethnic_Group**[Ethnic_Group_Other, Nationality], **Sports_Organisation**[Sports_Organisation_Other, Pro_Sports_Organisation], **Corporation**[Corporation_Other, Company, Company_Group], **Political_Organisation**[Political_Organisation_Other, Government, Political_Party, Cabinet, Military], **Location**[Location_Other, Spa, **GPE**[GPE_Other, City, County, Province, Country], **Region**[Region_Other, Continental_Region, Domestic_Region], **Geological_Region**[Geological_Region_Other, Mountain, Island, River, Lake, Sea, Bay], **Astral_Body**[Astral_Body_Other, Star, Planet, Constellation], **Address**[Address_Other, Postal_Address, Phone_Number, Email, URL]], **Facility**[Facility_Other, Facility_Part, **Archaeological_Place**[Archaeological_Place_Other, Tumulus], GOE, **Line**[GOE_Other, Public_Institution, School, Research_Institute, Market, Park, Sports_Facility, Museum, Zoo, Amusement_Park, Theater, Worship_Place, Car_Stop, Station, Airport, Port, Line_Other, Railroad, Road, Canal, Water_Route, Tunnel, Bridge]], **Product**[Product_Other, Material, Clothing, Money_Form, Drug, Weapon, Stock, Award, Decoration, Offence, Service, Class, Character, *ID_Number*, **Vehicle**[Vehicle_Other, Car, Train, Aircraft, Spaceship, Ship], **Food**[Food_Other, Dish], **Art**[Art_Other, Picture, Broadcast_Program, Movie, Show, Music, Book], **Printing**[Printing_Other, Newspaper, Magazine], **Doctrine_Method**[Doctrine_Method_Other, Culture, Religion, Academic, Sport, Style, Movement, Theory, Plan], **Rule**[Rule_Other, Treaty, Law], **Title**[Title_Other, Position_Vocation], **Language**[Language_Other, National_Language], **Unit**[Unit_Other, Currency]], **Event**[Event_Other, **Occasion**[Occasion_Other, Religious_Festival, Game, Conference], **Incident**[Incident_Other, War], **Natural_Phenomenon**[Natural_Phenomenon_Other, Natural_Disaster, Earthquake]], **Natural_Object**[Natural_Object_Other, Element, Compound, Mineral, **Living_Thing**[Living_Thing_Other, Fungus, Mollusc_Arthropod, Insect, Fish, Amphibia, Reptile, Bird, Mammal, Flora], **Living_Thing_Part**[Living_Thing_Part_Other, Animal_Part, Flora_Part]], **Disease**[Disease_Other, Animal_Disease], **Colour**[Colour_Other, Nature_Colour]]

Table 3.3: The approximate overlapping of different fine-grained tagsets

'Mapped to' coarse-grained classes MUC	CoNLL	ACE	Fine-grained classes	How many times overlapped?
-	MISC	FAC	Airport	4
LOC	LOC	LOC	City	4
LOC	LOC	LOC	Country	4
ORG	ORG	ORG	Government	4
PER	PER	PER	Person	4
-	MISC	-	Other	4
-	MISC	FAC	Bridge	3
LOC	LOC	LOC	Continent	3
LOC	LOC	LOC	Mountain	3
LOC	LOC	LOC	Region	3
LOC	LOC	LOC	River	3
-	MISC	-	Other	3
ORG	ORG	ORG	Educational	3
ORG	ORG	ORG	Religious	3
PER	PER	PER	Scientist	3
-	MISC	-	Book	3
-	MISC	WEA	Weapon	3
-	MISC	-	Game	2
-	MISC	-	War	2
-	MISC	FAC	Attraction	2
-	MISC	FAC	Building	2
-	MISC	FAC	Road	2
-	MISC	FAC	Station	2
LOC	LOC	GPE	City	2
LOC	LOC	GPE	Country	2
LOC	LOC	LOC	Border	2
LOC	LOC	LOC	Island	2
LOC	LOC	LOC	Other	2
LOC	LOC	LOC	Water	2
-	MISC	-	Animal	2
-	MISC	-	Email	2
-	MISC	FAC	Plant	2
-	MISC	-	URI	2
ORG	ORG	ORG	Corporation	2
ORG	ORG	ORG	Hotel	2
ORG	ORG	ORG	Museum	2
ORG	ORG	ORG	Other	2
ORG	ORG	ORG	Political	2
ORG	ORG	ORG	Sport	2
PER	PER	PER	Artist	2
PER	PER	PER	Group	2
PER	PER	PER	Individual	2
PER	PER	PER	Nationality	2
PER	PER	PER	Other	2
-	MISC	-	Drug	2
-	MISC	-	Food	2
-	MISC	-	Law	2
-	MISC	-	Painting	2
-	MISC	-	Play	2
-	MISC	-	Product	2
-	MISC	-	Sculpture	2
-	MISC	-	Show	2
-	MISC	VEH	Vehicle	2
-	MISC	WEA	Chemical	2
-	MISC	WEA	Nuclear	2

Approximate Overlapping of Different Fine-grained Tagsets The presented fine-grained tagset in Section 3.1.2.2 has a different variety in the term of the semantic classes involved. For all fine-grained classes presented in Table 3.2, there are 388 distinct fine-

grained classes. In Table 3.3 we tried to capture the overlapping between those tagsets in the fine-grained level. We found that, there are 55 classes have overlapped with at least two different fine-grained tagsets. For example, the 'Airport' class has been presented in four different tagsets while 'food' overlapped twice. The overlapping information presented in Table 3.3 helps to identify fine-grained classes that has been used in more than one tagset and therefore reflect their importance of presence in any created tagset.

In addition to the overlapping, we provided a mapping between the fine-grained classes into the three traditional coarse-grained tagsets. For example, the fine-grained 'Airport class is mapped into '-', 'MISC' and 'FAC' coarse-grained classes as suggested by MUC, CoNLL and ACE respectively. Moreover, this mapping shows that the coarse-grained traditional tagsets have a shortage of coverage. For example, classes such as 'Drug' and 'Food' have not equivalent mapping at the coarse-grained level.

3.1.3 Formal Definition of the Task of NER

To formally define the NER task, consider the following sentence in Figure 3.1 (the NEs are tagged by the surrounding [type]**NE**[type] symbols, where type represents the semantic class).

وفي عام ١٨١٨ دمرت [Population-Center] **الدرعية** [Population-Center] عاصمة [Nation] **الدولة السعودية الأولى** [Nation] على يد [Politician] **إبراهيم باشا** [Politician] ابن والي [Nation] **مصر** [Nation]، [Politician] **محمد علي باشا** [Politician]، لتنتهي بذلك [Nation] **الدولة السعودية** [Nation] إلا أن [Nation] **الدولة العثمانية** [Nation] لم تتمكن من البقاء طويلا في **نجد** [State-or-Province] بسبب استمرار الغارات ضد قواتها فانسحبت. [State-or-Province]

The English translation:
And in 1818 [Population-Center] **Diriyah** [Population-Center] the capital of [Nation] **the first Saudi State** [Nation] destroyed at the hands of [Politician] **Ibrahim Basha** [Politician], son of [Nation] **Egypt** [Nation] governor [Politician] **Muhammad Ali Basha** [Politician], putting an end to the [Nation] **Saudi State** [Nation], while that of the [Nation] **Ottoman Empire** [Nation] could not survive long in [State-or-Province] **Najd** [State-or-Province] because of the continuing attacks against its forces, so it withdrew.

Figure 3.1: An example of Arabic sentence having eight NEs

In this sentence, there are eight NEs each belonging to different semantic classes as seen in Table 3.4:

Table 3.4: The extracted NEs from the example sentence

NE	Gloss	Fine-grained class (type)
الدرعية	Diriyah	Population-Center
الدولة السعودية الأولى	First Saudi State	Nation
إبراهيم باشا	Ibrahim Basha	Politician
مصر	Egypt	Nation
محمد علي باشا	Muhammad Ali Basha	Politician
الدولة السعودية	Saudi State	Nation
الدولة العثمانية	Ottoman Empire	Nation
نجد	Najd	State-or-Province

The formal representation of the NE in this context is expressed as a sequence of tokens, as shown below:

$$\overbrace{Left\,Context}^{T_{(l-k)}...T_{(l-1)}} + \overbrace{NE}^{T_{(l)}...T_{(i)}...T_{(m)}} + \overbrace{Right\,Context}^{T_{(m+1)}...T_{(m+j)}}$$

where $T_{(i)}$ denotes a token and $l \leqslant i \leqslant m$.

The NE could span more than one token. Given any sequence of tokens, the first sub task is to successfully delimit the boundary of the NE by identifying both upper and lower boundaries, i.e. the first and last tokens, in the context. Once this is successfully achieved it is followed by the sub-task of classifying NE to appropriate semantic classes. Formally, let us assume that the NE is defined as a one chunk phrase and C represents the set of classes $c_1, c_2, ..., c_n$ where n is the total number of fine-grained classes. In this case the second sub-task is defined as the ability to predict the probability of the NEs belonging to class c_i. Then the highest probability assigned to class c_i is considered as being the target class that the NE statistically belongs to:

$$Class\,tag(t) = argmax\,P(NE|c_1^n)$$

3.1.4 Evaluation of NER

The evaluation of NER is an important step in ensuring the actual performance of such systems. It is generally undertaken by comparing the system's output with equivalent text already annotated by hand. Therefore, four common measures have been used; precision, recall, F-measure and the accuracy. The possibilities when evaluating the found entities and the entities that have not found are (Manning et al., 2008):

- True positive, an entity that was supposed to be found has been found.
- False negative, an entity that was supposed to be found was not found.
- False positive, no entity was supposed to be found, but one was found.
- True negative, no entity was supposed to be found, and none was found.

These cases can be made clear by using the following contingency table:

Table 3.5: The contingency table that show the four possibilities of finding named entities

	Correct	Not correct
Found	true positive (tp)	false positive (fp)
Not found	false negative (fn)	true negative (tn)

The precision is defined as the number of correct entities found divided by the total number of found entities.

$$Precision = \frac{correct\ and\ found\ entities}{found\ entities} = \frac{tp}{tp + fp}$$

The recall is defined as the number of correct entities found divided by the total number of correct entities.

$$Recall = \frac{correct\ and\ found\ entities}{correct\ entities} = \frac{tp}{tp + fn}$$

Using precision and recall alone will cause some problems in certain cases (Manning et al., 2008). For example, it is possible to get 100% recall by simply returning all possible entities while this does not evaluate the correctness of the retrieved entities. To overcome this issue, the weighted harmonic mean of both precision and recall can be calculated, and that is called the F-measure.

$$F_\beta = \frac{(\beta^2 + 1) * Precision * Recall}{(\beta^2 + Precision + Recall)}$$

where β adjusts the importance of recall over the precision. For example, having $\beta = 2$ results in weighting recall two times as precision. CoNLL uses $\beta = 1$ for their evaluation.

Another measurement that can be used is the accuracy, in other words, the number of correct results both found and not found divided by the total number of results. For named entity recognition, the accuracy is likely to be very high since the set of named entities in a text usually is rather small compared to the whole text. This results in a high accuracy since the set of true negatives will be much larger than the other three sets in contingency table.

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn}$$

For part-of-speech tagging and similar tasks, the accuracy works better as a measurement since every token should be tagged and all tags are equally important. This is not the case for NER, because the majority of the tokens in the corpus are tagged as 'O' (meaning not NE) and only 10% to 13% of the total corpus represents NEs (as will be seen in Section 5.3.1).

To closely analyse the error in tagging, a confusion matrix (sometimes called error matrix) is a common way to be used especially when the number of classes is greater than two (Manning et al., 2008). The confusion matrix shows for each pair of classes $< c1, c2 >$, how many NEs from class $< c1 >$ were incorrectly assigned to class $< c2 >$. In other

words, it is not only possible to see whether or not a tag is correct but also what kinds of mistakes are most common. For example as seen in Table 3.6, the NER system manages to distinguish the three traditional NE classes: person (PER), organisation (ORG) and location (LOC), but makes many errors within two classes. The confusion matrix can help pinpoint opportunities for improving the accuracy of the system. For example, to address the error in Table 3.6, we can see that the NER system struggled to distinguish between the classes ORG and LOC. One could introduce new set of features that can help the classifier to properly distinguish between those classes.

Table 3.6: An example of confusion matrix for a NER system that classify NEs into three classes (i.e. PER, ORG, and LOC)

	PER	ORG	LOC
PER	15	0	0
ORG	0	5	3
LOC	0	4	7

In practise, there are three different variations in performing the evaluation step, these being MUC, CoNLL and ACE.

MUC (Chinchor and Robinson, 1997) presents tolerant evaluation criteria in which the NER system is separately credited for the semantic tag and the textual boundary. The tag is considered correct whenever it matches the correct tag, regardless of the exact textual boundary and so long as there is an overlap. The textual boundary of the NE is considered correct whenever it matches the correct text, regardless of the tag.

On the other hand, CoNLL (Tjong Kim Sang and De Meulder, 2003) provides an exact match evaluation, where the entity is credited whenever it matches the tag and the textual boundary. Compared to the proposal by MUC, it offers strict criteria, but at the same time avoids boundary ambiguity. Precision, recall and harmonic F-measures are used to calculate performance. Precision measures the percentage of the correctly found NEs by the system. For example, if a system is able to detect and classify 20 NEs where 8 are correctly delimited and classified, then the precision is equal to 40%. Recall, on the other hand, measures the percentage of NEs appearing in the corpus that are accurately

found by the system. For example, if the system is able to correctly detect 8 NEs where there are a total of 10 NEs in the dataset, then the system's recall is 80%. The F-measure is used to evaluate the overall performance of a system.

ACE (2004) is a more complex evaluation method in comparison to MUC and CoNLL. It takes into consideration different parameters, including sub matching and coreference scoring. It evaluates the miss and the false alarm where the former counts the failure in capturing NEs, while the latter counts the error captured by the system. For each entity type, this method gives weight contributing to the final evaluation score. Due to the complexity of this evaluation method, the majority of Arabic NER studies instead use the CoNLL approach. These include Benajiba et al. (2009a); Abdul-Hamid and Darwish (2010); Darwish (2013).

3.2 Available Resources

Where groups of resources of different levels are required, it is essential to first establish the availability of linguistic resources in the creation of Arabic NER (Shaalan, 2013). This section therefore contains a review of the available Arabic linguistic resources in literature, including textual resources (i.e. annotated corpora and lexical resources) and a number of different tools (such as tokenisers, morphological analysers, POS taggers and environments to develop NER).

3.2.1 Corpora[1]

The collections of annotated textual data (i.e. corpora) are an important resource of reliable NER tasks (Nadeau and Sekine, 2007; Shaalan, 2013). The corpus should be of reasonable size and annotate every NE by delimiting its boundary and assigning to it the correct semantic tag. The majority of the available corpora focus on a single language (i.e. monolingual), such as ANERcorp (Benajiba et al., 2007).

Further researchers, such as Samy et al. (2005), have exploited the availability of par-

[1]This thesis uses the loose definition of the corpus as 'a text resource'

allel corpora (i.e. Spanish-Arabic) tagging the NEs in the Spanish corpus then projecting the result into the Arabic corpus by relying on the transliteration scheme.

Concerning the multilingual level, Mostefa et al. (2009) employ a semi-automatic approach to annotating the NE corpora of three languages: Arabic-English-French. The textual data has been collected from the Agence France Presse, covering the period between 2004 to 2006. The approach in the first step relies on a rule-based NE tagger called LIMA, which works by seeking triggers in the context such as (وزارة /wzArħ/ 'Ministry') and then examines the immediate right and left context to verify if the following phrase is NE. Manual inspection to establish the correctness of the automatic tagging is performed as the second step.

There are two trends of creating NE corpora covered by the literature. Firstly, there is the manual approach to creating such corpora, in which text from different sources is compiled and then (two or more) individuals recruited to manually annotate the NEs. Although this approach is considered as time consuming, the reliability of such corpora is assured by calculating the inter-annotator agreement between annotators, such as Kappa Stata (Carletta, 1996) or F-measure (Hripcsak and Rothschild, 2005; Zhang, 2013). The alternative trend is to develop the target corpus in an automatic manner. There are a number of different approaches towards achieving this goal. One advantage of this methodology is that it is time-saving (Zhang, 2013). The review of the literature covering both trends is discussed in the following sections.

3.2.1.1 Manually-created NE Corpora

There are a number of Arabic NE corpora that have been considered as gold-standard and used in studies such as those of Benajiba and Rosso (2008); Farber et al. (2008).

The majority of these corpora are governed by annual licenses, such as the ACE (2004) dataset. This is considered as a limitation for researchers with limited budgets for resources. Hence, in-house corpora have been developed, but with a limited domain and size, such as ANERcorp (Benajiba et al., 2007). However, the majority of such corpora are compiled from the newswire genre.

A newswire based corpus called ANERcorp is considered to be the earliest public-free corpus, and contains 170k tokens developed by Benajiba et al. (2007). ANERcorp follows the format proposed by CoNLL and annotates four NE types (PER, ORG, LOC and MISC). More recently, Mohit et al. (2012) has developed a smaller corpus of 74k tokens, called AQMAR, by drawing upon 28 Arabic Wikipedia articles. Although AQMAR follows the CoNLL tagset, the MISC class has been used to tag non-traditional tags, with MISC-0 being used to tag a generic miscellaneous entity. MISC-1 is used to tag the names of wars, particles, chemical elements and championships. MISC-2 is used to tag English entities, and the names of theories and prizes. Finally, MISC-3 is used to tag the names of computer components.

There are three NE corpora used extensively by researchers: ACE (2003, 2004, 2005). These corpora are all governed by LDC[2] and are inaccessible to the public. They follow the format suggested by an ACE tagset, capturing 7 and 45 coarse- and fine-grained types of NEs, respectively. These corpora have been widely used for Arabic NER, including (Farber et al., 2008; Benajiba et al., 2009a; Benajiba and Zitouni, 2009; Abdallah et al., 2012; Zitouni and Benajiba, 2014).

Table 3.7 summarises some aspects of the mentioned corpora.

[2]Linguistic Data Consortium: https://www.ldc.upenn.edu/

Table 3.7: List of available Arabic NE corpora (BN: Broadcast News; NW: Newswire; ATB: Arabic Tree Bank; and WB: Weblogs)

Corpus	ANERcorp	ACE2003	ACE2004	ACE2005	AQMAR
Publication date	2007	2003	2004	2005	2013
Licence	Free	LDC	LDC	LDC	Free
Genre	NW	BN, NW	BN, NW	BN, NW, WB	Wikipedia
Size	170k	55k	154k	104k	74k
Semantic classes	4 coarse	5 coarse	7 coarse and 45 fine	7 coarse and 45 fine	4 coarse
Following tagset	CoNLL	ACE	ACE	ACE	CoNLL

There are a number of further in-house corpora that have been used in Arabic NER, but these are unavailable for the public, including Nezda et al. (2006); Shaalan and Raza (2007); Mostefa et al. (2009); Elsebai (2009).

Recruiting humans to annotate textual data for particular NLP task is a critical for creating the required dataset to train a statistical classifier, but the annotation cost remains the issue. Recently, researchers have investigated a new methodology to overcome this obstacle by relying on crowdsourcing. Crowdsourcing, was firstly coined by Howe (2006), is the process of delegate particular task to a large group of people rather than to few trained annotators (Sabou et al., 2012; Hsueh et al., 2009). Crowdsourcing platforms such as Amazon Mechanical Turk (AMT)[3] and CrowdFlower[4] have allowed NLP researchers to develop their annotated corpora remotely by recruiting large number of annotators such as (Finin et al., 2010) to annotate NEs in Twitter data. Since those annotators are not trained, carefully designing the annotation task and evaluating the annotators quality is important to ensure the overall output quality of the developed corpus (Hsueh et al., 2009; Lease, 2011).

3.2.1.2 Automatically-created NE Corpora

The traditional approach to manually develop annotated corpus by recruiting a number of humans is a tedious and time-consuming. Instead, a promising trend in the research is to-

[3]https://www.mturk.com/mturk/welcome
[4]http://www.crowdflower.com/

wards automatically developing an annotated NE corpus beyond both traditional classes, and the newswire domain, in order to create novel resources. We mean by 'automatically-created' where there is no human intervention in the process of creating the annotated corpus. One of the earliest of these approaches is outlined by An et al. (2003), using the web to build the target corpus, with the aim of employing bootstrapping to build an annotated NE corpus from the web. Bootstrapping is the process of which one is given a small set of labelled data and a large set of unlabelled data, and the task is to induce a classifier (Abney, 2002). Hence initial instances of NEs from three traditional classes (i.e. PER, ORG and LOC), have been manually established. This is followed by the use of a search engine to fetch web pages using the searched entity in context. A sentence separator is used to detect the boundaries of the sentence and a set of heuristics is applied to filter the results.

A further approach utilises parallel corpora to build an NE corpus automatically. This relies on the suggestion that once one corpus is annotated, and then other, parallel, corpora can easily be annotated using projection. Ehrmann et al. (2011) have developed multilingual NE corpora for English, French, Spanish, German and Czech. The English corpus has been automatically tagged using an NE tagger and then a projection step applied to tag other sentence-level aligned corpora. A machine translation strategy has been used to translate the source NE into different languages, which is then reworked by applying a projection method (e.g. string matching). Similarly, Fu et al. (2011) have developed a Chinese annotated NE corpus exploiting the English aligned corpus, the difference being that the alignment is conducted between both corpora at the word-level.

Beyond the newswire-based corpora, Wikipedia becomes attractive for a number of different NLP tasks. Some researchers have exploited the unrestricted accessibility of Wikipedia in order to establish an automatic fully annotated NE corpus with different granularity. Meanwhile, others have been focusing solely on partially utilising Wikipedia to achieve specific goals, such as developing a NE gazetteer (Attia et al., 2010) or classifying Wikipedia articles into NE semantic classes (Saleh et al., 2010). It is crucial to

review efforts undertaken in this domain, since they are those most closely related to the current research.

Dakka and Cucerzan (2008) have presented the first work in which Wikipedia has been exploited for an NE task. Their goal was to classify Wikipedia articles into traditional NE semantic classes and a set of 800 random articles was manually annotated in order to use it with the classifier. Naïve Bayes (NB) and the Support Vector Machine (SVM) have been chosen as the statistical interface by exploiting a specific set of features, including bag-of-words, structured data, unigram and bigram context. More recently, Saleh et al. (2010) have proposed a similar approach to classifying multilingual Wikipedia articles into traditional NE classes. The assumption in this case is that the majority of Wikipedia articles relate to a NE and therefore sets of structured and unstructured data have been extracted in order to be used as a feature set when using an SVM. Among these features are bag-of-words, category links and infobox attributes. Thus, multilingual links are exploited in order to map classified articles for different languages.

Tkatchenko et al. (2011) expanded the classification into an 18 fine-grained taxonomy extracted from (BNN)[5]. In order to prepare training data for use in the classification stage, a small set of seeds is constructed (as undertaken by Nadeau et al. (2006)), in which a semi-supervised bootstrapping approach is developed, in order to construct long lists of entities in different fine-grained NE classes from the web. Once the list of entities has been constructed, the entities are then intersected with Wikipedia articles, in order to classify each article according to the target class. As a consequence, a set of 40 articles per fine-grained class has been produced, to be used as training with the NB and SVM. Several features have been selected in a way similar to (Saleh et al., 2010; Dakka and Cucerzan, 2008).

Rather than relying on machine learning, Richman and Schon (2008) have defined a set of heuristics involving the use of assigned category links to classify the article. Phrasal

[5]This is an annotated English NE corpus owned by LDC.
Available: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2005T33 [accessed 10 October 2014]

patterns for each semantic NE class were specified when a matching article was classified. Alternatively, the procedure searched the upper level of categories to find candidates. These articles are still classified according to traditional coarse-grained classes.

Nothman et al. (2008) presented the first attempt to transform Wikipedia into an annotated NE corpus for English. The assumption made was that many NEs are associated with the Wikipedia inter-links, i.e. the hyperlinks associated with a phrase in the context pointing to a separate article. Therefore, the procedure firstly requires identifying NEs by using heuristics to exploit capitalisation. Once NEs are identified, the second stage is to classify the target article into NE semantic classes. A bootstrapping approach is then used to extract seeds from a set of 1300 articles. For each article, two distinguished features are extracted; i.e. the head noun for the category links and the head noun for the definitional sentence. The corpus produced covered 60 fine-grained classes using two levels. An alternative approach to the same data set is presented by Tardif et al. (2009), in which the classification relies on supervised machine learning. Like Dakka and Cucerzan (2008), both NB and SVM have been used as statistical interfaces for the purposes of classification. A total of 2311 articles have been manually annotated and a combination of structured, and unstructured, features extracted.

Balasuriya et al. (2009) considers the complexity of NEs in Wikipedia as being far greater than those in the newswire corpus, and that it is therefore commonplace to evaluate an annotated corpus over similar texts in order to avoid domain transfer issues. A set of 145 random articles has been manually annotated according to CoNLL coarse-grained classes so as to evaluate the performance of a Wikipedia-based corpus reasonably. This evaluation demonstrates that the model trained on the automatically developed Wikipedia-based annotated corpus outperforms those trained on newswire gold-standard corpora.

3.2.2 Lexical Resources

A further important resource is comprised of lexical resources (i.e. a gazetteer). Some researchers refer to these as 'white lists' or 'dictionaries' (Shaalan and Raza, 2007, 2008). Regardless of the terminology, lexical resources consist of a list of NEs tagged with an appropriate NE class, and they serve as external knowledge where it can be used in rule-based NER systems as a matching component (Shaalan and Raza, 2009), as a feature in supervised ML approaches (Benajiba et al., 2009a), or as seeds for semi-supervised NER approaches (Althobaiti et al., 2013).

It is important to develop such resources for Arabic, and therefore researchers tend to develop their own resources by different means. Benajiba et al. (2007) has manually compiled (through the use of web resources) a list of NEs of traditional NE classes. This consists of 1920, 262, and 1950 personal, organisational and locational entities. A further attempt has been undertaken by Shaalan and Raza (2009), in which a collection of NEs were compiled from different resources (such as the ACE2005 and Arabic Treebank (ATB) corpora and other websites resources). This resource contains a considerable number of entities, i.e. 263598, 273491 and 4900 personal, organisational and locational entities. However, this form of resource is unavailable to the public. The CJK Dictionary Institution[6] provides a considerable collection of Arabic personal names associated with English transliteration. This resource is governed under license, which prevents its use by researchers.

Attia et al. (2010) have developed a means of building a NE lexicon for Arabic, whereby Wikipedia is used alongside Arabic WordNet. This resource focuses solely on personal and locational NEs of the size 16038 and 4588, respectively. Alkhalifa and Rodrguez (2009) have presented an approach to enrich the Arabic WordNet by relying on the Arabic Wikipedia as an external knowledge source. Similar to Attia et al. (2010), Ehrmann et al. (2011) have developed a multilingual resource in which substantial lists of personal and

[6]The CJK Dictionary Institute, Inc. http://www.cjk.org/cjk/arabic/arabsam.htm

organisational names (i.e. 205k) have been created from daily news. The use of Wikipedia has combined to retrieve different varieties of the way the names are expressed.

'NoorGazet' is a domain specific personal-only gazetteer created by Bidhendi et al. (2012) in which 88k of NE entries have been gathered manually. The aim of Bidhendi et al. (2012) is to develop a NER system for Arabic ancient Islamic text and thus the developed gazetteer reflects this domain. The NEs have been extracted from an Islamic book named (جامع الحديث /jAmʕ AlHdyθ/ 'Jamee Alhadith') and a tokenisation step has been applied to separate the full name into parts. The final distinct single-word NEs consist of 18238 personal names.

3.2.3 Environments and Tools for NER

There are a number of language-independent environments and tools in the literature providing a suite to develop reliable NER systems by employing either a list of rules or statistical models. In this section, a review of such environment and tool are presented in relation to Arabic NER.

3.2.3.1 Rule-based Environments for NER

GATE (Cunningham, 2002): This is an open source software, which works as an environment supporting a large number of languages, including Arabic. It is shipped with different components, e.g. tokeniser; POS tagger; gazetteers, etc. In relation to NER, it facilitates creating a set of rules that detect NEs by relying on the Java Annotation Patterns Engine (JAPE), which is built based on regular expressions. GATE has been used for a number of Arabic NER studies, including the following: Elsebai et al. (2009); Elsebai and Meziane (2011); Abdallah et al. (2012).

LingPipe (Alias-i., 2008): This is a Java-based toolkit that facilitates the implementation of different NLP tasks, such as POS, NER and spelling correction. It is built to be language independent, and was therefore employed by AbdelRahman et al. (2010); Zaghouani (2012), by training the ANERcorp to be evaluated with his proposed approach. This platform is governed by different licenses, including for the purposes of research.

NooJ (Silberztein, 2005): This is a development environment that supports creating a rule-based NER system (such as GATE) but with limited functionalities. NooJ uses dictionaries and grammars to perform morphological and syntactical analysis. Nooj has been used in limited studies of Arabic NER, including in Mesfar (2007); Fehri et al. (2011).

3.2.3.2 Supervised ML toolkits for NER

An effective approach is the implementation of NER systems by relying on supervised models, requiring an annotated dataset. There are number of ML optimised implementations of different probabilistic model in the literature, which have been widely used in the community of different languages. Yet Another Small MaxEnt Toolkit (YASMET) is a C++ generic toolkit of the Maximum Entropy (ME) model. This has been used by a number of studies, including Benajiba and Rosso (2007); Benajiba et al. (2007). For Support Vector Machines (SVM), Yet Another Multipurpose CHunk Annotator (YamCha[7]) is an open source tool designed to perform sequence labelling tasks, such as NER, POS and phrase chunking. This has been used by Benajiba et al. (2008a,b).

There have been a number of implementations related to Conditional Random Fields (CRF). CRF++[8] is an implementation widely used for Arabic NER (Benajiba and Rosso, 2008; Abdul-Hamid and Darwish, 2010; Darwish, 2013; Morsi and Rafea, 2013). CRFsuite[9] has received the same attention as CRF++, due to the fact that it facilitates modification of the features generating code. It has been extensively used in English NLP tasks (e.g. Turian et al. (2010)) and in Arabic (e.g. Abdul-Hamid and Darwish (2010);

[7]http://chasen.org/~taku/software/yamcha/
[8]https://code.google.com/p/crfpp/
[9]http://www.chokkan.org/software/crfsuite/

Darwish (2013)). Lavergne et al. (2010) have recently developed a rapid toolkit known as 'Wapiti' for sequence labelling tasks, and which implements ME, ME-HMM and CRF within one toolkit. It favours other toolkits, in that it requires less time for training steps compared with (for example) CRF++. It has been employed in a number of different sequence labelling tasks in English, including Nouvel et al. (2012); Bodnari et al. (2013).

3.2.4 Basic Preprocessing Tools for Arabic

As a morphological complex language, Arabic has attracted the attention of a number of researchers aiming to develop tools to address several aspects of processing the language, including the morphology analyser and POS tagger. This section comprises a brief discussion regarding well-known pre-processing tools with a direct relationship to the task of Arabic NER.

Buckwalter Arabic Morphological Analyser (BAMA) (Buckwalter, 2002): This is one of the most widely used tools for Arabic morphology analysis. It consists of three components: lexicon, compatibility tables, and an analysis engine. The lexicon consists of words and lemmas, and dictionaries of prefixes, stems and suffixes. The compatibility tables stores items of prefix-stem, stem-suffix and prefix-suffix to ensure compatibility. For example, the prefix (و /w/ 'and') is compatible with all stems of the type 'noun (NN and NNS)'. Each stem is associated with an English translation as a gloss. This feature allows several studies in Arabic NER to utilise the capitalisation of the glossed word (e.g. Farber et al. (2008)). The analysis engine is the third component of BAMA where (for a given input) it returns all possible analyses of the input as an output, without the ability to select the correct one based on the context. The transliteration feature of the output facilitate the readability of the results, particularly for those lacking in an ability to read

Arabic scripts. BAMA has been used in a number of Arabic NER studies (e.g. Farber et al. (2008); Elsebai and Meziane (2011); Al-Jumaily et al. (2012)).

Morphology Analysis and Disambiguation for Arabic (MADA) (Habash et al., 2009): This consists of two components: the morphology analyser and the disambiguation. The morphology analyser component relies on BAMA, where it derives its strengths by producing information concerning the analysed word, including stem, diacritisation and POS tagging of approximately 500 different tagsets. However, it shares the BAMA's limitations in the case of no given analysis by BAMA as an out-of-vocabulary. MADA is shipped with a flexible tokenisation feature, which permits users to specify their chosen type of tokenisation. The second component is the disambiguation, where SVM models have been applied to select the correct output from BAMA, based on the context. Benajiba et al. (2008a,b, 2010) have extensively used MADA to extract Arabic morphological features to be used in NER.

AMIRA (Diab, 2009): This is a set of tools including a tokeniser, POS tagger and Base Phrase Chunker (BPC) (i.e. shallow syntax parser). AMIRA is a successor to the ASVMTool (Diab et al., 2007). In contrast to MADA, AMIRA has no dependency on deep morphological knowledge to perform the analysis, instead relying on SVM as sequence models to learn generalisation. The tokenisation component (known as AMIRA-TOK) relies on the segmentation knowledge provided by the Penn Arabic Treebank (PATB) and deals with tokenisation as a character-level chunking. As a second component, the POS is called AMIRA-POS. It uses a set of 72 tagset, an extended version of Reduced Tag Set (RTS) (Habash, 2010, p. 80), known as the Extended Reduced Tag Set (ERTS)[10]. AMIRA-BPC is the third component and works by grouping a sequence of words into phrases, such as NPs and VPs. It attempts to produce the longest possible sequence of words to form phrases. AMIRA-BPC covers several types of phrases presented in Table 3.8. Both POS and BPC information have been utilised as features in Arabic NER, as investigated by Benajiba et al. (2010); Koulali and Meziane (2012)

[10]Full description of ERTS is presented at the beginning of and used throughout this thesis in Table 3

Table 3.8: Types of phrases used by AMIRA

Phrase	Description	Arabic example	Translation
ADJP	Adjectival phrase	جيداً	Well
ADVP	Adverbial phrase	سريعاً	Quickly
CONJP	Conjunctive phrase	و الفلسطينيون	And the Palestinians
PP	Prepositional phrase	خلال الحفلة	During the party
PREDP	Predicative phrase	إن المطر غزير	The rain is copious
PRTP	Particle phrase	لا سيما	Not as long
NP	Noun phrase	الزفاف الجماعي	The group wedding
SBAR	Subjunctive construction phrase	الذي يدخل أولاً	That enters first
INTJ	Interjective phrase	يا أخت	Oh sister
VP	Verb phrase	يأكل الولد طعامه	The boy eats his meal

3.3 Approaches to Arabic NER

Unlike languages in which NER has reached maturity (e.g. English and French) Arabic NER has only recently begun to attract researchers. Nadeau and Sekine (2007) conducted a survey classifying the NER approaches into two main streams: handcrafted rules and those that are machine-learning (ML) based. ML in itself is classified into supervised, semi-supervised and unsupervised ML. Approaches that have been encountered for Arabic NER in the course of this research are reviewed below.

3.3.1 Handcrafted Rule Based NER

An early contribution to Arabic NER has been made by Maloney and Niv (1998). This involved a combination of a morphological analyser and a pattern recognition engine, the former being responsible for identifying the start and end of a token, and the latter for identifying the corresponding applied pattern. This effort was focused on specific semantic classes, i.e. PER, LOC, number and time. The evaluation was made by randomly selecting portions of textual data from the Al-Hayat newswire. The authors reported 89.5% precision, 80.8% recall and 85% F-measure.

Abuleil (2004) has developed an NE tagger for QA systems with the aim of eventually acquiring a database of names by utilising keywords and specific verbs to identify potential NE. It captures triggers such as (الدكتور /Aldktwr/ 'Dr.') and assumes that NEs should appear no further than three words away from the trigger. Moreover, the NEs should be no longer than 7 words. Then the directed graph can be used to draw a relationship between words contextualised within phrases. Finally, the verification step is accomplished by applying rules to the names. This approach was evaluated in over 500 articles drawn from the Al-Raya newspaper, where the system scored 90.4%, 93% and 92.3% for precision, recall and F-measure, respectively.

Mesfar (2007) has utilised the NooJ environment to develop NER for Arabic, with the types identified being PER, ORG, LOC and currency. The proposed approach consists of three components in a pipeline structure, the components being tokeniser, morphological analyser and NER tagger. It was evaluated over a newswire-based contextual dataset with the results being reported per class, with the F-measure for location names being 76%.

Shaalan and Raza (2007) have compiled a large lexicon list dedicated to personal names, forming a gazetteer, extracted from a number of different resources. The gazetteer contained over 472000 entries, including first, middle and last names, job titles and country names. Regular expression rules were applied to identify the availability of personal names in context. Given that Arabic is a highly inflectional language, and has relatively free word ordering, designing generic hand-crafted rules is challenging. Traboulsi (2009) partially utilised contextual clues to identify personal names, identifying triggers such as a variety of verbs, including (قال /qAl/ 'Said') and (أخبر /Oxbr/ 'Tell') , as keywords preceding a personal name.

Elsebai et al. (2009); Elsebai and Meziane (2011), in contrast to Mesfar (2007), used GATE to develop a rule-based system for personal Arabic names. Instead of relying on a large gazetteer of personal names, Elsebai et al. (2009) merged parts of speech with manually created keywords and heuristic rules. An Arabic morphological analyser (i.e.

BAMA, Buckwalter (2002)), was used to extract features integrated with the rules in conjunction with two forms of lexical keywords: such as verbal (e.g. (قال /qAl/ 'Said')) and title-identification triggers (e.g. (دكتور /dktwr/ 'Dr.')). Two experiments were conducted using 700 and 500 news articles, with the F-measure of both experiments being 89%.

Al-Shalabi et al. (2009) present a similar approach to Abuleil (2004); Elsebai et al. (2009); Elsebai and Meziane (2011), in which they rely on identifying triggers (i.e. keywords and certain verbs) within the context, assuming that they will be followed by proper nouns. A set of rules has been prepared to assist in identifying the location and type of the NE. This approach has been evaluated, as follows: over 20 documents were selected randomly from Al-Raya[11] and Alrai[12] newspapers from which the detected proper nouns were categorised into: person; organisation; location; scientific; temporal; equipment; and events where the overall precision was 86.1%.

A slightly wider granular NER, with the ability to identify ten different types of NEs, was proposed later by Shaalan and Raza (2009), This extended the work of Shaalan and Raza (2007), which relied on gazetteers and lists of rules derived from large resources. A disambiguation method was employed to solve the inevitability of lexical overlap. Shaalan and Raza (2007) developed a system focused specifically on personal names (called PERA), whereas in NERA (Shaalan and Raza, 2008, 2009) the semantic coverage was extended to include an increased number of classes, i.e. person; location; organisation; date; time; ISBN; price; measurement; phone numbers; filenames.

Shihadeh and Günter (2012) developed a system known as ARNE, which is reliant on

[11]http://www.raya.com
[12]http://www.alrai.com

a gazetteer developed by Benajiba et al. (2007) (i.e. ANERgazet) to classify the detected NEs into traditional classes, i.e. person, organisation and location. The system goes through a preprocessing step, in which it performs tokenisation, transliteration and POS tagging. Since this approach relies solely on the gazetteer list, its performance is low, scoring a 30% F-measure.

A real time NER system for Arabic has been proposed by Al-Jumaily et al. (2012). This approach consists of a number of components working together to identify three traditional semantic classes, with lists of prefixes, morphological variation and gazetteer dictionaries being prepared. The gazetteer was compiled from ANERgazet, GATE and DBpedia[13] to increase the coverage. Its main purpose is to tokenise the input text, followed by undertaking several steps to identify its pattern, and then a lookup step to verify its availability in the gazetteer. This approach is evaluated over the ANERcorp. The personal, locational and organisational F-measures were 77.27%, 70.87% and 57.30%, respectively.

In a similar fashion to approaches reliant on local grammars (such as Traboulsi (2009)), Zaghouani et al. (2010) adapted the Europ Media Monitor (EMM) platform by introducing three components, i.e. preprocessing, lookup full names and local grammar to recognise unknown names. The main difference between this approach and others relying on grammar consists of the fact that (where applicable) Zaghouani et al. (2010) maintain the use of language-independent rules, in conjunction with those that are language-dependent. The evaluation was made over a corpus compiled and annotated from the Assabah and Alanwar newspapers[14]. The overall F-measure achieved was 74.95%. Two years after this initial approach, the same system was named RENAR, and was evaluated over the ANERcorp dataset. The aim of this evaluation is to compare the performance of this method with other machine learning based studies (such as Benajiba and Rosso (2007)). RENAR outperforms other systems for locational NEs, scoring an F-measure of 87.63%.

[13]DBpedia is a project that aim to extract structured information from the Wikipedia to be available to users http://dbpedia.org

[14]http://www.alanwar.com/

3.3.2 Machine-learning Based NER

The majority of machine learning based approaches are supervised, in that the machine learns from an annotated corpus and attempts to predict unseen text. However, semi-supervised and hybrid methods have recently received increased attention (Althobaiti et al., 2013; AbdelRahman et al., 2010).

Researchers have addressed NER using supervised ML as sequence labelling in the same manner as the POS and text chunking. The performance of any supervised ML approach is affected by two important components: (1) the probabilistic model and (2) feature engineering. Both will be discussed in detail in this section.

3.3.2.1 Probabilistic Models

There are a number of probabilistic models in the literature that have been utilised in developing NER including: Maximum Entropy (ME); Structured Perceptrons (SP); Decision Tree (DT); Support Vector Machines (SVM); and Conditional Random Fields (CRF) (Nadeau and Sekine, 2007; Shaalan, 2013). More recently, an Arabic NER system has been developed by Mohammed and Omar (2012), based on the Neural Network (NN). In this current thesis, a brief background is presented of two models (i.e. ME and CRF) because they are the state-of-the-art of the baseline methods in the literature for NER (Benajiba et al., 2010; Zirikly and Diab, 2014).

Maximum Entropy (ME): Sequential prediction has been utilised in multiple NLP tasks, including NER, given a sequence of tokens as input $x = (x_1, ..., x_n)$ and a predefined set of labels $y = (y_1, ..., y_c)$, where c is the number of labels (i.e. classes). The task is to find the most effective sequence of labels with the highest conditional probability among all possible label sequences.

The highest probability assigned to label y_i is then considered as being the target class to which the token x_i statistically belongs:

$$y_i = argmax\ p(y_1^c|x_i)$$

denoting that X and Y are to be the space of the possible inputs and output variables, respectively. The output of the classification process can be represented as a function $h : X \rightarrow Y$. The state feature functions $f_k(y_i, x_i)$, where $k = 1, ..., m$ are used to represent facts in relation to the observations. For example, a state feature function could represent the token itself as a feature, such as:

$$f_k(y_i, x_i) = \begin{cases} 1, & if\ x_i =' London'\ and\ y_i = LOC \\ 0, & Otherwise \end{cases}$$

The ME sequence tagging formula is as follows:

$$p(y_i|x_i) = \frac{1}{Z(x)} exp\left(\sum_{k=1}^{m} \lambda f_k(y_i, x_i)\right) \tag{3.3.1}$$

where $Z(x)$ is a normalisation function defined as:

$$Z(x) = \sum_{y \in Y} exp\left(\sum_{k=1}^{m} \lambda f_k(y_i, x_i)\right) \tag{3.3.2}$$

Conditional Random Fields (CRF): CRF as a discriminative undirected graph model is frequently employed for sequence labelling, where there is an important relationship between adjacent inputs, i.e. two adjacent words. CRF is widely used in different fields, such as: NLP tasks (Morsi and Rafea, 2013; Darwish and Gao, 2014), computer vision (He et al., 2004) and biomedical identification (Settles, 2004).

In addition to the state feature function previously defined, transition feature function $g_k(y_{i-1}, y_i, x_i)$ is used to represent the observation sequence and labels at different positions of i. For example, a transition feature function could represent the token itself as a feature whilst considering adjacent label, such as:

$$g_k(y_{i-1}, y_i, x_i) = \begin{cases} 1, & if\ x_i =' London'\ and\ y_i = LOC\ and\ y_{i-1} = Other \\ 0, & Otherwise \end{cases}$$

The CRF is defined as:

$$p(y|x) = \frac{1}{Z(x)} exp(\sum_{i=1}^{n}\sum_{k=1}^{m}\lambda_k g_k(y_{i-1}, y_i, x_i)) \tag{3.3.3}$$

Where Z(x) is a normalisation function defined as:

$$Z(x) = \sum_{y \in Y} exp(\sum_{i=1}^{n}\sum_{k=1}^{m}\lambda_k g_k(y_{i-1}, y_i, x_i)) \tag{3.3.4}$$

3.3.2.2 Feature Engineering

Feature engineering is a process in which characteristics, properties or knowledge of the words are extracted and fed to the probabilistic model as input. This provides an abstraction representation of the input, where the features can be Boolean values such as 'is capitalised?', a numerical expression such as 'the length of the token', or a nominal string such as 'the first three characters'.

The most important aspect affecting the reliability and accuracy of supervised ML based NER is the set of features employed; these are either language-independent or language-dependent. In the case of language-independent features, a variety of lexical and contextual features have emerged. On the other hand, morphological and syntactical features have received most of the attention at the lexical level, as well as parts of speech. This section will review both types of features.

Language-independent Features: Lexical, contextual and external knowledge features are those represented as language-independent features where they are not specifically related to a specific language. Lexical features is a set of features extracted from the actual surface of a token (e.g. the first or last three letters) (Abdul-Hamid and Darwish, 2010). Contextual features are those related to the context, such as the position of the token in a sentence. The set of external knowledge features is injected to the vector space by outsourcing, e.g. gazetteers.

a. Lexical Features: One of the earliest approaches to extract lexical features in Arabic NER has been conducted by Benajiba et al. (2007). The author uses the current token as a feature, as well as filtering the tokens if they are stop words by assigning a Boolean flag. Benajiba et al. (2008a,b) have employed the same feature. Benajiba et al. (2009b,a); Benajiba and Zitouni (2009) utilised orthographical features by extracting the n-gram of six characters. The reason for extracting the prefixes and suffixes is in order to capture the clearly represented prefixes and suffixes in the statistical model in a heuristic manner. Abdallah et al. (2012) limit the length of the prefixes and suffixes to 2. Koulali and Meziane (2012) explicitly distinguish the location and extract the length of the token as feature. As previously discussed, Arabic, unlike English, has no capitalisation feature. However, Benajiba et al. (2008a, 2010) have been able to derive an assumed capitalisation for Arabic word by utilising the MADA tool, which provides a set of morphological knowledge and a gloss translation for the analysed word. This feature assumes that a token is a proper noun if the translated gloss begins with capitalisation.

b. Contextual Features: Consideration of the contextual features in the statistical model is utilised by the insight that the NEs appear and share specific context, with the surrounding words being the most frequently employed contextual features. The window size of the surrounding words varies between -/+1 to -/+5. Benajiba et al. (2009a) consider the optimal size of the window to be -/+1, whereas Benajiba and Zitouni (2009) confirm that -/+2 is sufficient, if appropriate tokenisation is performed to separate prefixes and suffixes. Statistical-based features (such as the t-test and mutual information between the current token and its surroundings) have been calculated by Abdul-Hamid and Darwish (2010).

c. External Knowledge: External knowledge varies, depending on its actual role in the NER systems. An obvious source of knowledge is the gazetteer, where lists of NEs are collected, either manually (Benajiba et al., 2007; Bidhendi et al., 2012) or compiled in an automatic manner (such as Nothman et al. (2008) for English). The gazetteer has

been used extensively by a number of studies, including those of Darwish and Gao (2014); Darwish (2013); Koulali and Meziane (2012). The positive effect of injecting such features depends on capacity, where a reasonably larger size implies improved coverage. A number of studies (such as Benajiba et al. (2009b, 2008a)) have employed the ANERgazet. The Arabic Wikipedia was used to enrich ANERgazet with an increased number of entities (Benajiba et al., 2009a,b; Koulali and Meziane, 2012). Moreover, lexical triggers have also been used to provide clues to predict the presence of NEs (Algahtani, 2012).

Language-dependent Features: Arabic contains language-dependent features and characteristics. A number of researchers have investigated the effect of exploiting such features to develop Arabic NER (i.e. Benajiba et al. (2008a)), with POS (among others) being used extensively (Zitouni and Benajiba, 2014; Morsi and Rafea, 2013; Zitouni and Benajiba, 2014). The intuition behind using POS-based features is the expectation that NEs are analysed as either common nouns (NN or NNS) or proper nouns (NNP). Benajiba and Rosso (2007); Farber et al. (2008) demonstrate improvement by including the POS features. However, the prediction task is complicated by the fact that NE can appear anywhere in the sentence and therefore a shallow syntactical feature (i.e. base phrase chunk (BPC)), has been used to overcome this situation, assuming NE is present in nominal phrases (NP) (Zitouni and Benajiba, 2014). Koulali and Meziane (2012); Benajiba et al. (2009b) have used an AMIRA toolkit to analyse a sentence and then extract the BPCs.

Deeper language-dependent features have been examined by Benajiba et al. (2008b); Benajiba and Rosso (2008) in which the morphological features (such as aspect, person, definiteness, gender and number) have been exploited in the statistical model by relying on MADA to analyse the Arabic sentence and then extract the morphological features. The usefulness of such features has been established.

3.3.2.3 State-of-the-art Supervised ML Arabic NER

Different supervised Machine Learning (ML) probabilistic models have been used in developing Arabic NER with a traditional set of semantic classes. Each study was centred

mainly on four tuples: (1) the semantic tagset; (2) the probabilistic model; (3) a set of features; (4) the genera of the dataset. Here, a review of each contribution will be reported in relation to those tuples.

There has been an early exploitation of Maximum Entropy (ME) in developing Arabic NER systems. One early approach to exploit such a classifier was presented by Zitouni et al. (2005); Nezda et al. (2006) was to examine the importance of the morphological stemming on Arabic NER. It began by segmenting the Arabic text by relying on a semi-supervised method similar to Lee et al. (2003), in order to enclitic the prefixes and suffixes off the stem. ACE2003 and ACE2004 corpora were used in the experiment, through extracting the lexical, contextual and shallow syntactical features. The experiment demonstrates that (including the stem information in the model) improves the performance by 2.9% F-measure.

Nezda et al. (2006) developed a NER system called CICEROARABIC, with the ability to detect and classify 18 different NE types. Seven of these are used to tag nominal NE (such as PER, ORG and LOC), while the rest are used to tag numerical expressions (such as percent, age and date, etc.). A textual corpus of 800k of tokens compiled from the Penn Arabic Treebank (PATB) and the Prague Arabic Dependency Treebank (PADT) (Hajic et al., 2004) was manually annotated to conduct the experiment by relying on the ME similar to that of Zitouni et al. (2005). The input text went over a number of steps, commencing from tokenisation by relying on the light8 stemmer presented by Larkey et al. (2002). This was followed by the extraction of a traditional set of features, including actual and stemmed words, prefix and suffix, and contextual features. The corpus was then divided into training and test datasets of 75% and 25%, respectively, and then fed to the ME classifier. The reported overall F-measure was 85.51%.

Benajiba et al. (2007) developed an in-house corpus known as ANERcorp, following CoNLL guidance, and a gazetteer known as ANERgazet. Both the corpus and the gazetteers were compiled from a newswire domain. Those resources have been widely employed by other researchers (Abdul-Hamid and Darwish, 2010). Benajiba et al. (2007)

developed a NER system called ANERsys by learning an ME classifier through relying solely on traditional lexical, contextual and gazetteer features. In a subsequent work, (Benajiba and Rosso, 2007) exploited language dependent features (i.e. POS and BPC) to overcome the limitation of ANERsys where it fails to tag multi-words NEs. In this approach, two separate steps have been introduced. The first step is dedicated to delimiting the boundary of the NE phrase, whereas the second step is used to assign tag to the detected phrases. The performance of the system was boosted from 55.23% to 65.91% F_1. This demonstrates that the language dependent features are important in the development of efficient NER. One year later, instead of having two separate steps, Benajiba and Rosso (2008) investigated the application of a CRF probabilistic model by using the same corpus and feature set in single model, as in Benajiba and Rosso (2007). The reported F-measure score was 79.21%. This also confirms that selecting an appropriate probabilistic model, as well as the correct features set, will enhance the overall performance of NER.

A cross-corpora study has been conducted by Benajiba et al. (2008a), in which SVM has been used as a probabilistic model over ANERcorp, ACE2003, ACE2004 and ACE2005 datasets. As with Benajiba et al. (2007); Benajiba and Rosso (2007, 2008), lexical, contextual and gazetteer features were used. Moreover, a new set of morphological features have been extracted and injected into the model. The most successful F-measure scored was 82.71% over ACE2003.

Benajiba et al. (2008b) investigated the most effective set of features for each semantic class. SVM and CRF were used to classify each class independently by using several sets of features. A voting scheme was advised to rank the features based on the output performance from either classifier. An incremental-based approach was applied, in which a new set of features were added on the top of the one optimised, in order to reach the highest performance possible. The best performance was 83.5% over ACE2003. Morsi and Rafea (2013) conducted incremental experiments in a similar manner to Benajiba et al. (2008b), in order to evaluate a set of 14 extracted lexical and contextual features as well as the POS. The best reported F-measure result was 68.05%, using the ANERcorp

corpus.

In order to confirm the importance of language dependent features in Arabic NER, more elaborate work have been presented in Benajiba et al. (2009b,a). Benajiba et al. (2009b) employed SVM as classifier; however, Benajiba et al. (2009a) also recruited ME and CRF. The aim of both studies is to establish the appropriate classifier and the set of features leading to superior performance; the best reported being 83.34% F-measure over ACE2003. Benajiba et al. (2009a) significantly conclude that SVM reveals improved performance over CRF in the cases involving a small number of features, whereas CRF performs better in large feature spaces. Koulali and Meziane (2012) have presented a similar approach to Benajiba et al. (2009b), using an SVM classifier in conjunction with a pattern extractor component. A new set of features (such as a 'rare word' feature) has been used by compiling a list of the less frequent words in the dataset (i.e. appear less than 10 times). A word will be assigned a binary flag if belongs to this list. ANERcorp was used to perform the experiment with the resulting F-measure score being 83.2%.

The work of Benajiba et al. (2009a) has been used as a baseline model for a new study presented by Benajiba et al. (2010), in which the head word as a syntactic feature has been exploited. Due to coverage of the extracted instances of the syntactic features being small, a bootstrapping mechanism has been advised in order to enrich the training data by syntactic features. An English NER tagger proposed by Zitouni and Florian (2009) has been used in order to tag the English text of a parallel Arabic-English aligned corpus, followed by a projection step. This approach was examined over three datasets (ACE2003, ACE2004 and ACE2005) and by use of the ACE coarse-grained tagset. The best F-measure score was 84.32 over ACE2003.

Farber et al. (2008) relied on the morphological analyser (i.e. MADA) and the Structured Perceptron (SP), as proposed by Collins (2002), in order to develop NER. They reported that the morphological features extracted from MADA improved the overall system. They emphasised the ability of MADA to assign an English gloss next to analysed Arabic words. Hence, if the gloss word is capitalised, the gloss has been used as a feature

alongside capitalisation. This approach has been focused on person, organisation and geo-political NEs and tested over the ACE2005 dataset, with the overall F-measure being 75.7%.

Abdul-Hamid and Darwish (2010) have examined an approach reliant on a simplified set of features, including the leading and trailing character and word n-grams and word length. The aim of the character n-gram is to capture linguistic clues such as (ال /Al/ 'the') which gives an insight into the proper name of family names. In this study, no external knowledge (i.e. a gazetteer) has been involved. ANERcorp has been employed to evaluate the proposed approach where the 'MISC' class has not been included in the experiment, and the reported F-measure was 81%.

Bidhendi et al. (2012) examined Noor, an NER system based on CRF used to detect and classify personal NEs from ancient Islamic Arabic text. Since the system is dedicated to Islamic textual data, a corpus and gazetteer have been created, known as NoorCorp and NoorGazet. NoorCorp has been compiled from three genera: (1) a historic book; (2) traditional prophetical narrations; (3) a jurisprudential book. For the training phase, in addition to using similar traditional lexical and contextual features (as presented by Benajiba et al. (2007); Benajiba and Rosso (2008); Benajiba et al. (2008b,a)), an AMIRA tool was used to tokenise and produce POS for each token as language dependent, similar to those in (Benajiba et al., 2008b,a). The highest reported F-measure was 99.93%. However, important experimental details have not yet been released, including the size of training and test portions.

Unlike the majority of previous studies, Mohit et al. (2012) investigated the detecting of NEs within a broader domain: Arabic Wikipedia. The traditional tagset (i.e. person, organisation and location), as well as the MISC category, were used to tag the NEs. Since

there was no available annotated dataset for the Arabic Wikipedia, they proposed a semi-supervised approach to build their corpus. However, a small portion of gold-standard text has been manually annotated to serve as test data. They used the structured perceptron, (described in (Collins, 2002)) as a probabilistic model, and extracted a set of lexical, contextual and morphological features. The key aim of this approach is to apply a cost function penalising the recall error, since poor recall is one of the most serious issues in transferred domain NER. The small dataset annotated manually was used to test this method where the reported result revealed an 8% improvement on F-measures when cost function was applied and in comparison with the baseline. However, the proposed approach revealed degradation of precision.

Darwish and Gao (2014) investigated the issue of NER on microblogging sites, i.e. Arabic tweets. The baseline model relies on CRF and adopts a similar feature set to that found in (Abdul-Hamid and Darwish, 2010). A set of 5069 tweets was manually annotated using the three traditional tagset presented by MUC. The newswire-based corpus (i.e. ANERcorp) has also been used as a training dataset, in order to evaluate the performance (regardless of whether the training data was from the same genera). Since classifying tweets by using newswire-based corpus yields a low F-measure (i.e. 29.9%) a semi-supervised approach was proposed, with the aim of developing a gazetteer from unlabelled tweets by first tagging the input and then applying a weighting mechanism to ensure minimisation of the noisy results. The resulting twitter-based gazetteer has been used as an external feature in the CRF model. The experiment reveals that the overall F-measure reaches 65.2%.

Due to the fact that Arabic lacks significant orthographical signs found in English (e.g. capitalisation), cross-lingual mapping facilitates an exploitation of such features. Darwish (2013) has proposed two approaches towards Arabic-English derived features. The first approach consists of a reliance on the ontology presented by DBpedia, and the cross-language links between Arabic and English Wikipedia pages. The second approach is a reliance on a machine translation framework called Moses (Koehn et al., 2007), which

generates phrase translations for Arabic sentences. CRF has been used as a probabilistic model and the features in the baseline model are similar to those proposed by Benajiba et al. (2008b); Abdul-Hamid and Darwish (2010). In addition to ANERcorp, two new corpora have been manually created. The first corpus is 15k of newswire-based tokens compiled from the RSS feed of the Arabic version of the Google news service, and named as NEWS. The second corpus is comprised of 26k of tokens drawn from Twitter, i.e. TWEETS. Applying the derived capitalisation features in the probabilistic model yields improvement across corpora where the F-measure for ANERcorp, NEWS and TWEETS has been risen by 4.4%, 9.4%, and 6.8%, respectively.

Mohammed and Omar (2012) have proposed an approach using Artificial Neural Networks (ANN) to detect and classify the four traditional CoNLL based tagsets. Tokenisation and transliteration steps have been undertaken prior to performing the classification. The features mentioned are a set of triggers per class to be used as cues. An in-house newswire-based corpus of 150k tokens was developed and used in the experiment. The reported overall F-measure was exceptional, scoring 92.36%. It is important to mention that, the ANN works well on a small number of classes and features and may well not be extendible to 50 classes.

3.3.3 Hybrid Based NER

Hybrid approaches aim to exploit the usefulness of approaches such as the rule-based and the statistical models in a single system. The earliest attempt was undertaken by AbdelRahman et al. (2010), who devised an algorithm to extract a set of patterns for each NE class (i.e. ten classes). A predefined confidence threshold is specified, based on the number of occurrences of the pattern. The pattern-based features are injected with others that are both language-dependent and independent by training a CRF classifier. The results were reported per class instead of the overall performance where (for example) the F-measure of the personal class scored 67.80%.

The rule-based method proposed by Shaalan and Raza (2008) has been exploited by

Abdallah et al. (2012) to form a new hybrid method by integrating with the statistical model. GATE has been used to re-implement the rules. The statistical model was based on the Decision Tree (DT). The hybrid system was implemented in a pipeline structure, in which the rule-based component runs first to tag the text. The output of this stage is compiled into the feature space, with other language-dependent and independent features. The set of features was then fed into the DT in order to predict the three traditional NE classes (i.e. PER, ORG and LOC). The approach was evaluated over the ACE 2003 and ANERcorp corpora.

Shaalan and Oudah (2014) undertook a further investigation, in which the number of NE classes was expanded from 3 to 11 types, including new classes such as time, measurement and price. The goal of this study is to evaluate the integration of a number of statistical models, including DT, SVM and LR. This approach has been evaluated over ACE2003, ACE2004 and ANERcorp.

Like Abdallah et al. (2012), Zayed and El-Beltagy (2012) have developed a hybrid NER that focuses only on personal names and employs an increased number of language-dependent features (i.e. morphological). The performance of this approach over ANERcorp dataset is exceptional, with the F-measure scoring 94.5%.

The hybrid approaches mentioned in this section have utilised extracted patterns by relying on a set of handcrafted rules from the text and then inducing those patterns in the statistical classifier. The two crucial issues are: extracting a set of rules requires linguistic knowledge of a particular domain; generalising these rules to avoid the over-fitting problem. Therefore, in this thesis we instead rely on supervised machine learning and evaluate this method across domains (i.e. newswire and Wikipedia).

Chapter Summary

A comprehensive literature review of Arabic NER has been presented in this chapter. The approaches to Arabic NER vary from hand-crafted rules to machine learning-based. On rule-based approaches, the work presented by Zaghouani (2012) is salient because it relies

on local grammar to extract NEs. On supervised machine learning approaches, the series of work conducted by Benajiba et al. (2009b,a, 2008a, 2010) has the advantage comparing with the other. The authors comprehensively studied the effect of extracted morphological knowledge from Arabic words and injected that knowledge into the classifier as features. They achieve high performance as in (Benajiba et al., 2010) where they scored 84.32% F_1. Nevertheless, all efforts have concentrated on very limited semantic classes, i.e. coarse-grained. This set of classes is not enough nowadays to upper-level applications such as Question Answering (QA) and Ontology Construction (OC). Moreover, all those efforts have been applied to newswire domain. Instead, this thesis pushes the research in Arabic NER steps forward in three dimensions by:

1. Proposing a hierarchy-based taxonomy of two levels to represent the semantic classes.

2. Presenting a methodology to develop scalable resources, i.e. gazetteer and corpora, in automatic manner.

3. Investigating novel approaches to represent the features that go beyond the window and sentence boundaries.

Before proceeding into the discussion about the implementation of the fine-grained NER for Arabic, two important resources need to be implemented. Therefore, in Chapter 4 and Chapter 5, the advised approach of developing required resources for Arabic fine-grained NER (i.e. gazetteer and corpora) will be covered in detail.

Part III

FINE-GRAINED RESOURCE CREATION

This part addresses the first research question concerning the building of fine-grained named entity resources. Chapter 4 discusses the methodology devised in order to create scalable gazetteer, while Chapter 5 discuss the approach of developing annotated corpora from different sources to be used as training dataset.

Chapter 4

Developing Scalable Fine-grained Gazetteer

Chapter Synopsis

In the previous two chapters, the necessary background was presented. In Chapter 2, an introduction to the Arabic language and some aspects of Arabic NE were given. Chapter 3 comprehensively reviewed the literature of Arabic NER.

Since the supervised ML approach is selected to develop fine-grained NER for Arabic, lexical (i.e. gazetteer) and textual (i.e. corpora) resources need to be developed. Therefore, in this chapter we will discuss our approach toward developing scalable fine-grained gazetteer. We address this issue by mapping the Arabic Wikipedia (AW) into a predefined set of semantic NE classes (i.e. 50 fine-grained classes). This task is formalised as a document classification problem where the AW articles represent the documents, and the fine-grained NE tagset are the target classes. We use several supervised ML classifiers to perform this task in a controlled experiment in the following sequence:

- Defining the semantic fine-grained NE tagset to be used (see Section 4.1).
- Since this approach relies on supervised ML, a reasonable size of annotated training dataset is required, and this is presented in Section 4.2.

- Two important issues affected the classification process, i.e. feature representation and engineering, and those will be discussed in Section 4.3 and 4.4.

As a result, a fine-grained gazetteer of Arabic, called WikiFANE$_{Gazet}$, is developed of the size of 68355 NEs.

4.1 Defining Fine-grained Semantic NE Tagset

Whether to define fine-grained semantic NE tagset by inventing a new taxonomy or utilising an existing one is an important decision. Different NER studies offer different tagsets; i.e. heuristic taxonomies. For example, Sekine et al. (2002) relied on WordNet to develop a suitable hierarchy-based tagset, in addition to analysing sets of questions used in a Text Retrieval Conference TREC-QA task. The ACE forum defined its two-level taxonomy to extract the most important NE from the textual data. Brunstein (2002) designed a two-level taxonomy to annotate the answer types of questions directly related to the NER task. Other tagsets, such as the one presented by Balasuriya et al. (2009) were adopted from Sekine et al. (2002) and Brunstein (2002). From these examples, it is evident that specifying the goal when designing such a taxonomy is crucial, and should be clearly stated. For example, if an NER system is applied to biomedical data, it would be wrong to add fine-grained semantic classes about 'Person', but the category 'Gene' would be important.

Therefore, the characteristics of the tagset we define are centred on the following:

1. Generic and wide enough (i.e. fine-grained): to be useful for Wikipedia and newswire domains and with no more than 50 classes, due to the limitation of our hardware specifications when performing the experiments;

2. Easy to be mapped back into coarse-grained tagset, i.e. compatibility with MUC, CoNLL and coarse-grained ACE; and

3. Compatible to an extent with fine-grained ACE tagsets, because the most used

Arabic NER corpora follow the ACE tagset, i.e. ACE2003, ACE2004, ACE2005. (More details about the available corpora were presented in Section 3.2)

It is evident that there is no widely agreed fine grained taxonomy that can be directly adopted for Arabic; although the ACE taxonomy is a reasonable choice in the sense that it organises granularity into two layers, i.e. coarse- and fine-grained. In the evaluation of ACE (2005), the number of fine grain classes is 45. This taxonomy is designed in two levels of granularities and frequently used in the newswire domain. Moreover, a two-level taxonomy allows us to map a tagset into different traditional schemes easily, such as CoNLL or MUC.

Thus, the ACE (2005) tagset was selected to form the basis of our fine-grained tagset. Since ACE is originally designed for a newswire domain we applied some amendments to tailor it for use with a relatively open domain corpus, such as Wikipedia. For example, there are many articles in Wikipedia about people in different subclasses, such as scientists, athletes, artists, politicians, etc. These fine-grained classes are not included in ACE, as it only includes three sub-classes: the individual, group and indeterminate. In our tagset, we divided the 'Person' class into the following fine-grained classes: Politician, Athlete, Businessperson, Artist, Scientist, Police, Religious, Engineer, Group. An additional modification was performed; i.e. a new class called 'Product' was added which involved the following fine-grained classes: Book, Movie, Sound, Hardware, Software, Food, Drug. The tagset that is used for this research is presented in Table 4.1.

4.2 Document Annotation; Strategy and Evaluation

The annotation of the set of AW articles is required in order to train a classifier to perform the document classification process. Therefore, two Arabic native speakers were involved in the annotation process, using the NEs tagset presented in Section 4.1. To decide the total number of articles to annotate, we reviewed a similar task for English. We found that, the total number of annotated articles varies from 800 articles as in (Nothman et al., 2008) to 4936 as suggested by Saleh et al. (2010). For this thesis, we allocated two

Table 4.1: The two-levels fine-grained tagset used in this research.

Coarse-grained Classes	Fine-grained Classes
PER: Person	Politician, Athlete, Businessperson, Artist, Scientist, Police, Religious, Engineer, Group.
ORG: Organisation	Government, Non-Governmental, Commercial, Educational, Media, Religious, Sports, Medical-Science, Entertainment.
LOC: Location	Water-Body, Celestial, Land-Region-Natural.
GPE: Geo-Political	Continent, Nation, State-or-Province, County-or-District, Population-Center, GPE-Cluster.
FAC: Facility	Building-Grounds, Subarea-Facility, Path, Airport, Plant.
VEH: Vehicle	Land, Air, Water.
WEA: Weapon	Blunt, Exploding, Sharp, Chemical, Biological, Shooting, Projectile, Nuclear.
PRO:Product	Book, Movie, Sound, Hardware, Software, Food, Drug.

dedicated weeks for the annotators to perform this task. We found that the annotators can annotate 300 documents per day. Therefore, it was decided that a reasonable goal would be to annotate 4,000 documents[1] and the annotators used a self-developed annotation tool to facilitate the annotation process (as seen in Figure 4.1) and both annotators were given guidelines, which clearly defined the distinguishing features of each class, including a practical method to pursue the annotation. The annotators were initially given the first 500 articles to annotate as a training session, in order to evaluate and identify limitations that might then be expected to manifest during the annotation process. It was expected that there would be a lower level of agreement between them in this round. In order to evaluate the inter-annotator agreement between the annotators we used the Kappa Statistic (Carletta, 1996). The overall annotation task, including the training session, was divided into three cycles to ensure the resolution of any difficulties the annotators might encounter. After each cycle, the Kappa was calculated and reported.

Table 4.2 shows that the overall inter-annotator agreement was calculated for different

[1]The 4000 articles have been selected randomly

Figure 4.1: Annotation tool used to annotate the Wikipedia articles

sizes of documents, i.e. 500, 2000 and 4000. This revealed difficulties that might be encountered during the annotation process.

Table 4.2: The overall inter-annotator agreement

Level	Kappa : n=500	Kappa : n=2000	Kappa : n=4000
Coarse-grained	92	98	99
Fine-grained	80	95	97

It is found that, the percentage of the coverage of the articles referring to NEs in the annotated documents is 74%.

4.3 Feature Representation

Feature representation affected the way the classification process was modelled in order to classify given Arabic Wikipedia (AW) articles and to then produce the mapped NE class for this article; otherwise the article would not relate to a NE. In this research, we conducted a comprehensive investigation to evaluate different methods of representing features in order to evaluate those most suitable to our task.

- **Term Presence (TP):** For each given document, the feature representation was

simply counted by examining the presence of the tokens in the document. There was no consideration given regarding the frequency of the tokens.

- **Term Frequency (TF):** This represents how many times the tokens in our corpus were found in a given document.

  For a given set of documents $D = d_1, d_2, ..., d_n$ where n is the number of documents. The term frequency (TF) for a given token (t) is calculated thus

  $$TF(t, D) = \sum_{d \in D} frequency(d, t)$$

- **Term Frequency-Inverse Document Frequency (TF-IDF):** This reveals how important a given token is to a document within the corpus. It involves scaling down the most frequent words across the documents while scaling up rare ones. The (TF-IDF) is then calculated by multiplying the term frequency (TF) with the inverse document frequency (IDF) as follows:

  $$TF - IDF(t) = TF(t, d) \times IDF(t)$$

  where:

  $$IDF(t) = log \frac{|D|}{1+|\{d:t \in d\}|}$$

  where $|d : t \in d|$ is the number of documents the term (t) appears in.

4.4 Feature Engineering

The nature of AW articles differs when compared with traditional newswire documents, as newswire articles have a tendency to be of a particular length and size due to certain externally imposed conditions. This does not apply to AW, and so some articles are very short while others are very long. Therefore, this necessitates a careful extraction of the most useful textual elements that offer a good representation of the article. We believe that using complete tokens in articles contributed surplus noisy data to the model. Therefore, we manually investigated several AW articles of different types in order to define appropriate locations. We decided to compile our raw dataset based on four different locations, based on specific aspects of the AW articles. These are the articles title **(t)**,

the first sentence **(f)**, category links **(c)** and infobox parameters **(p)** (see Figure 4.2 for an illustrated example).

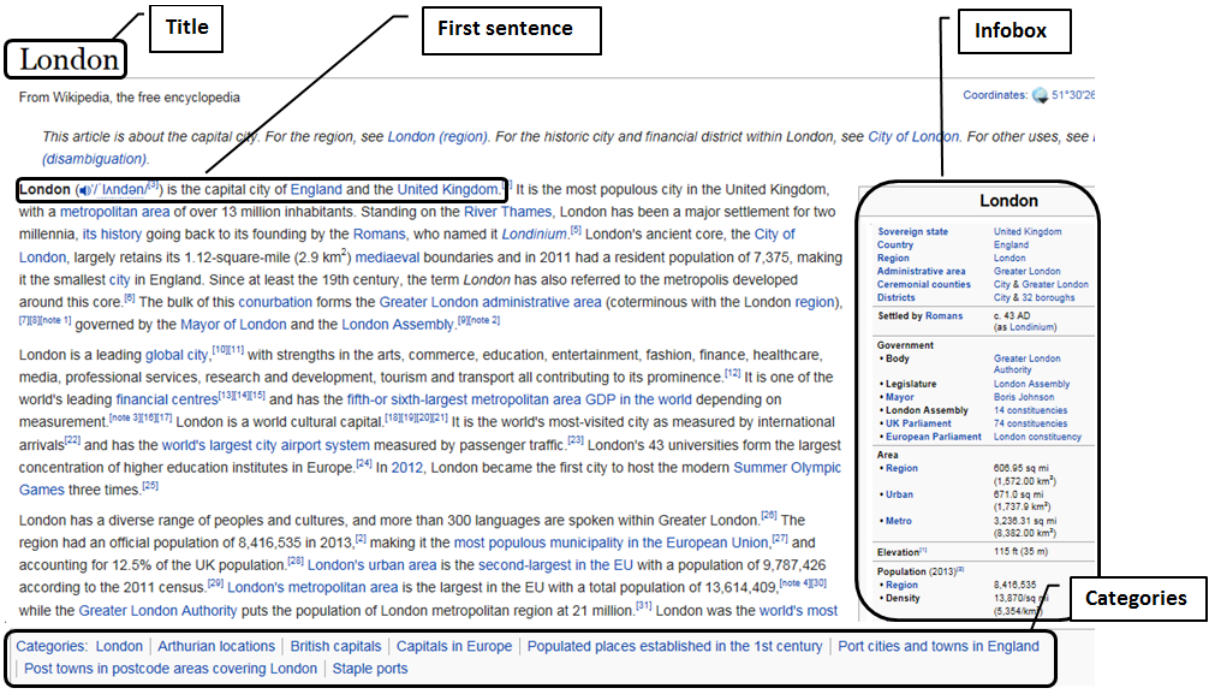

Figure 4.2: An example showing different locations of Wikipedia article

Although the dataset was modelled as a bag-of-words, we were interested in investigating the optimum feature set used within this representation, so as to yield the highest performance for the classification process. The feature sets presented below either involve eliminating or augmenting data, i.e. features, which have been defined as either language-dependent or independent:

- **Simple Features (SF):** This represents the raw dataset as a simple bag of words without further processing. The idea in this case is to evaluate the nature of the full word representation of the AW articles in this task.

- **Filtered Features (FF):** In this version, the following heuristics have been applied in order to obtain a filtered version of the dataset:

  1. Removing the punctuation and symbols (none-alphabetical tokens).

2. Filtering stop words[2].

3. Normalising digits where each number has been converted into a letter (*d*). If we have a date such as 1995, this will be normalised to *dddd*.

- **Language-dependent Features (LF):** The stemming is the term used to describe the process that reduces all words with the same stem to a common form, usually by stripping each word of its derivational and inflectional suffixes (Lovins, 1968). In this feature, we aim to investigate the effect of using the stem instead of the full word form to avoid data sparseness. We relied on the NLTK::ISRIStemmer package (Bird et al., 2009) which is based on the algorithm proposed by Taghva et al. (2005).

- **Enhanced Language-dependent Features (ELF):** This feature set was processed in several steps, which are explained below:

  1. Tokenising all tokens within the data set using the AMIRA tokeniser developed by Diab (2009) by applying the tokenisation scheme of (conj+prep+suff)[3] instead of stemming.

  2. Using the same tool to assign the part of speech (POS) for each token would allow filtering of the dataset by involving only nouns (for instance) in the classifier.

  3. Isolation of tokens based on their locations: this is a novel idea for representing the dataset. The intent in this case being to isolate similar tokens, which appear in different locations on a given document. The intuition behind this is that some tokens that appear in a particular location, i.e. title, first sentence, categories and infobox, of the AW articles, are more discriminative in certain locations rather than the whole article. The idea with isolation would be to attach to each token an identifier, i.e. **(t)** for title, **(f)** for first sentence, **(c)** for category and **(i)** for infobox, to act as a header based on the location in

[2]We relied on an extended list of stop words. Available at: http://arabicstopwords.sourceforge.net/
[3]In this scheme the conjunctions, prepositions and suffixes are separated by white space.

Arabic example:
t_القوات t_الجوية t_المصرية
f_في f_العسكري f_الطيران f_فرع f_هي f_المصرية f_الجوية f_القوات f_
f_القوات f_المسلحة f_المصرية f_.
c_عربية c_جوية c_قوات c_مصر c_في c_الطيران c_المصرية c_الجوية c_القوات
i_الدولة i_تاريخ i_قائد i_الانشاء i_القوات i_الجوية i_المصرية i_المعارك i_الشرفية

English translation:
t_Egyptian t_Air t_Force
f_Egyptian f_Air f_Force f_is f_the f_military f_aviation f_branch f_of
f_the f_Egyptian f_armed f_forces f_.
c_Egyptian c_Air c_Force c_Aviation c_in c_Egypt c_Arab c_air c_forces
i_State i_Created i_Date i_Egyptian i_Air i_Force i_commander i_Honorary i_battles

Figure 4.3: The isolated representation of the article titled 'Egyptian Air Force'

which the token appears. An example of the results of the isolation process are shown in Figure 4.3.

In this case example, the feature representation of the token (المصرية /AlmSry/ 'The Egyptian') presented in the first sentence does not affect, and is not affected by, the same token in the category links or title, even though they have identical glyphs. Surprisingly, the implementation of this idea contributed significant improvements to the classification process.

4. For term presence (TP) only, we applied the most informative features for the top 1000 informative features. To calculate the most informative features we used a Chi Square test (Yang and Pedersen, 1997).

4.5 A Pilot Experiment at the Coarse-grained Level

Classifying AW into coarse-grained tagset is conducted as a pilot experiment in order to learn the best practice (i.e. including the feature representation and engineering, and the best-performed classifier) that can be applied into the fine-grained tagset (as will be seen in Section 4.6). Therefore, we started the experiments by splitting the annotated dataset into training and test sets of 80% and 20% respectively[4]. To the best of our knowledge, there is no similar comparable work for the target language and dataset; therefore we will instead analyse our findings as comprising a comparative study of several properties. The experiment was designed to evaluate three factors; the feature representation, feature sets and the probabilistic models. Therefore we extensively use this 3-touple representation to facilitate analysis of the results.

Several text classifiers were applied in order to evaluate performance: Naïve Bayes (NB), Multinomial Naïve Bayes (MNB), linear Support Vector Machine (SVM) and Logistic Regression (LR). The experimentation was conducted relying on both Scikit-learn (Pedregosa et al., 2011) and NLTK (Bird et al., 2009).

Since the traditional NB classifier relies on term presence we started by evaluating those factors alone. Table 4.3 presents the feature sets used, in conjunction with three standard metrics, i.e. Precision, Recall and balanced F-measure (i.e. F_1[5]).

Table 4.3: The classification results when using Naive Bayes across different feature sets where (TP) is applied. (The bold style represents the highest result per metric)

Classifier	Feature set	P	R	F
NB	SF	60	54	56
	FF	**62**	62	62
	LF	59	69	63
	ELF	**62**	**81**	**70**

[4]The random selection and splitting has been applied to ensure the quality of the training and test sets

[5]Throughout this thesis, when F-measure is mentioned it means F_1

Although both FF and ELF have scored identical points in precision, ELF shows significant improvements in the recall and F-measure. This gives the impression that, the enhanced features, i.e. ELF, have boosted the model so as to recall more documents.

To ensure that the results are statistical significant, we applied Cochran's T test (Cochran, 1950). Cochran's T test is a non-parametric statistical test to verify whether k algorithms have identical results. The null hypothesis and alternate hypothesis are described below respectively:

H_0: Different approaches presented in Table 4.3, have the same result and error rate, and there is no significant statistical difference between them.

H_a: Different approaches presented in Table 4.3, have the different results and error rate, and there is significant statistical difference between them.

The Cochran's T test statistic is:

$$T = k(k-1)\frac{\sum_{j=1}^{k}(X_j - \frac{N}{k})^2}{\sum_{i=1}^{b} X_i(k - X_i)}$$

where

k is the number of experiments

X_j is the column total for the j^{th} experiment

b is the number of blocks

X_i is the row total for the i^{th} block

N is the grand total

By applying the Cochran's T test, we get the value of $T = 168.2$, and the probability (P-value) $p < 0.001$. Since the resulted T value is much greater than the P-value, thus we can safely reject the null hypothesis as the difference between algorithms are highly statistically significant.

Table 4.4 shows the result when applying the remaining classifiers in the case of the TF as the feature representing the backbone.

Table 4.4: The classification results using MNB, LR and SVM over different feature sets where (TF) is applied

Feature set	MNB			LR			SVM		
	P	R	F	P	R	F	P	R	F
SF	82	82	81	81	79	77	86	**87**	86
FF	82	82	82	87	87	87	**87**	86	86
LF	77	76	76	83	83	83	83	83	83
ELF	**88**	**88**	**88**	**88**	**88**	**88**	**87**	**87**	**87**

The tuples {TF, ELF, LR} and {TF, ELF, MNB} achieved the best result of all the metrics where they scored 88% on F-measure. {TF, SF, SVM} has proven to perform very well (scoring 86% F-measure) by merely using a simple feature set. Moreover, using ELF leads to the highest performance across all classifiers.

The results of applying TF-IDF for features representation are shown in Table 4.5.

Table 4.5: The classification results when using MNB, LR and SVM over different feature sets where (TF-IDF) is applied

Feature set	MNB			LR			SVM		
	P	R	F	P	R	F	P	R	F
SF	85	85	85	89	89	89	89	**89**	**89**
FF	86	86	85	**90**	**90**	**90**	**90**	**89**	**89**
LF	79	78	78	86	86	86	85	85	85
ELF	**88**	**88**	**88**	89	89	89	89	89	**89**

In the main, all classifiers showed improvements when TF-IDF is applied. The tuple {TF-IDF, FF, LR} outperforms all other models where this shows the ability for LR to generalise the optimum model in order to achieve the highest performance. {TF-IDF, FF, SVM} scored 90% on precision, while both the recall and F-measures scored 89%.

4.6 Fine-grained Document Classification Results

After conducting several experiments to classify AW into coarse-grained NE tagset as shown in 4.5, we use the best practice in term of classifiers and features representation to pursue the fine-grained classification. Therefore, we decided to learn both SVM and LR as

probabilistic models and the TF-IDF as feature representation. Moreover, we decided to evaluate the effect of bigram representation of the features in comparison with unigram.

Table 4.6 shows the overall results for the fine-grained classification. There are three main findings. First, both classifiers tend to perform in a very similar way; therefore, in practice, use of either classifier to perform the final classification for the whole Wikipedia dataset will be expected to deliver very similar results. The second finding is that the bigram features have little effect when different features are set. Finally, the highest result for both classifiers was achieved using the 'ELF$_{Uni}$' feature.

Table 4.6: The average fine-grained classification results when using LR and SVM over different feature sets where (TF-IDF) is applied

Feature set	SVM			LR		
	P	R	F	P	R	F
SF$_{Uni}$	78	79	78	78	79	78
SF$_{Uni+Bigram}$	80	81	80	80	81	79
FF$_{Uni}$	80	81	80	81	82	80
FF$_{Uni+Bigram}$	81	82	81	81	82	81
LF$_{Uni}$	77	78	77	78	79	78
LF$_{Uni+Bigram}$	79	80	79	79	80	79
ELF$_{Uni}$	**82**	**83**	**82**	**82**	**83**	**82**
ELF$_{Uni+Bigram}$	81	82	81	**82**	82	81

4.7 Introducing a Fine-grained Arabic NE Gazetteer

We used the set of 400 annotated articles as training data in order to classify all AW articles using SVM. The result of this classification is the development of scalable fine-grained NE gazetteer named WikiFANE$_{Gazet}$. The developed gazetteer consists of 68355 entities and has a coverage of 50 fine-grained classes (according to the tagset presented in Section 4.1). Based on our best knowledge, the only Arabic NE gazetteer currently available is that produced by Benajiba et al. (2007) covering only three traditional NE classes, i.e. PER, ORG and LOC. The size of this gazetteer is 4132 entities. Table 4.8 compares the distribution between ANERgazet and WikiFANE$_{Gazet}$ in the coarse-grained level. The distribution of the fine-grained classes is presented in Table 4.9. It is clearly

shown that, WikiFANE$_{Gazet}$ has superiority in the sense of type and coverage where WikiFANE$_{Gazet}$ is 16 times larger than ANERgazet. The gazetteer produced is freely available to the research community to use and extend[6].

The Quality of the Gazetteer: The performance of document classification across all Wikipedia articles is crucial to avoid error propagation from the document classification stage when compiling the final version of the annotated corpus. Therefore, this evaluation focused on this aspect. After classifying all articles to the target NE classes, we drew another 4000 articles, to be represented as a sample for all Wikipedia articles, and manually annotated them. The selection of the articles was made by selecting the first 4000 articles with identical glyphs to those used most frequently in other Wikipedia articles. This criterion ensured that the most frequent NE was classified properly with a minimum error rate. After this, we calculated the inter-annotation agreement between the manually annotated and the automatically classified documents. Table 4.7 shows the result for both levels of granularity. The overall Kappa for the fine-grained level is 82.6%, and this is consistent with the results shown in Section 4.6. This shows that the error rate is at a minimum, even when performing the classification across all Wikipedia articles with small amounts of training data.

Table 4.7: Inter-annotation agreement between the classified articles and the gold-standard

Level	Accuracy	Overall Kappa
Coarse-grained	85.8	84.02
Fine-grained	82.9	82.6

4.8 Chapter Summary

In this chapter we tackled the problem of mapping AW articles into a predefined set of NEs classes in order to develop scalable fine-grained gazetteer. We modelled this problem as a document classification task and comprehensive experiments were empirically conducted

[6]The fine-grained Arabic NE gazeetter WikiFANE$_{Gazet}$ is freely available at http://www.cs.bham.ac.uk/~fsa081/resources.html

Table 4.8: The distribution of NEs for different gazetteers across coarse-grained NE classes

Class	ANERgazet	WikiFANE$_{Gazet}$
PER	1920	31123
ORG	262	6664
LOC	1950	1424
GPE	NA	20494
FAC	NA	2182
VEH	NA	521
WEA	NA	279
PRO	NA	5668
Total	4132	68355

Table 4.9: The distribution of NEs for WikiFANE$_{Gazet}$ across fine-grained NE classes

Class	# of entities	Class	# of entities
PER:PERSON	**31123**	**FAC:FACILITY**	**2182**
Artist	9475	Airport	194
Athlete	6648	Building-Grounds	1643
Businessperson	198	Path	282
Engineer	171	Plant	3
Group	1453	Subarea-Facility	60
Police	410	**VEH:VEHICLE**	**521**
Politician	6008	Air	219
Religious	4890	Land	228
Scientist	1870	Water	74
ORG:ORGANISATION	**6664**	**WEA:WEAPON**	**279**
Commercial	1309	Blunt	6
Educational	1069	Chemical	10
Entertainment	166	Exploding	92
Government	691	Nuclear	61
Media	772	Projectile	48
Medical-Science	115	Sharp	25
Non-Governmental	899	Shooting	32
Religious	157	Biological	5
Sports	1486	**PRO:PRODUCT**	**5668**
LOC:LOCATION	**1424**	Book	974
Celestial	262	Drug	112
Land-Region-Natural	538	Food	319
Water-Body	624	Hardware	422
GPE:Geo-Political	**20494**	Movie	2570
Continent	13	Software	1038
County-or-District	1093	Sound	233
GPE-Cluster	311		
Nation	1370		
Population-Center	16361		
State-or-Province	1346		

in order to evaluate several properties concerning the classification task. The output of this chapter is the automatic development of fine-grained NE gazetteer for Arabic of the size 68355 entities which can be exploited to develop the fine-grained NER system as will be seen in Chapter 6.

Developing scalable gazetteer automatically from Arabic Wikipedia is an approachable methodology. An important step to perform this task is to study the underlying structure of Wikipedia carefully in order to utilise it toward achieving this goal. The idea used in this chapter to apply machine learning technique, i.e. document classification, facilitates the automatic creation of such resource. In this sense, two important issues should be carefully designed.

First is the way the features have been represented in which three possible features representations that were investigated: Term Presence (TP) simply counts the presence of the tokens in the document; Term Frequency (TF) represents how many times the tokens are found in a corpus; and, Term Frequency-Inverse Document Frequency (TF-IDF) reveals how important a given token is to a document within the corpus.

The second issue is the set of features that were involved in the classification. Four different sets of features have been investigated in this chapter: Simple Features (SF) represent the raw dataset, as a simple bag of words without further processing; Filtered Features (FF) represent the dataset after several filtering steps have been taken (including the removal of punctuation, stop words and normalising digits), Language-dependent Features (LF) report the usefulness of the stem representation of the token; and, Enhanced Language-dependent Features (ELF) represent linguistic features, including tokenisation, assigning the POS to each token and distinguishing the tokens based on their location on the Wikipedia page.

In this chapter, we conclude that both classifiers, i.e. LR and SVM, have performed almost similar in this task. This turn the attention to focus on features representation and engineering instead of applying new classifier algorithm.

In the next chapter, we will discuss our approach to develop fine-grained NE corpora

to be used as training data.

Chapter 5

Developing Fine-grained Training Data

Chapter Synopsis

In the previous chapter, we presented our approach to develop a fine-grained gazetteer (i.e. lexical resource) by exploiting the richness of Arabic Wikipedia. In this chapter, we will discuss our approach toward developing fine-grained NE corpora in two ways. The first way, as will be shown in Section 5.1, develops these resources in an automatic manner by exploiting the underlying structure of the Arabic Wikipedia. Section 5.2 will present the development of the gold-standard corpora from two different genres. After developing the corpora, it is possible to study the nature of the Arabic NEs within the context. Therefore, we conducted a series of corpus-based evaluations to compare some of the characteristics of the corpora. The comparison, which will be shown in Section 5.3 involves studying the tag density and uniqueness; the distribution of length; and phrases and semantic classes.

5.1 Automatically Developing a Scalable Dataset

5.1.1 Wikipedia as a Source of Data

Wikipedia has been selected as a backbone knowledge source to develop the annotated corpus. Several reasons behind the selection of this resource are expressed below:

1. Semi-structured data: A careful inspection of the underlying structure of Wikipedia shows that the sort of knowledge it includes is not unstructured textual data, such as that elsewhere on the web. More details of the structure of Wikipedia will be presented in Section 5.1.2.

2. Publically accessible: Cost and the restricted access policy are considered as barriers to prevent researches from tackling such problems. Alternatively, Wikipedia provides a daily archive of all content which can be downloaded in the form of a dump of XML data. This sort of resource can be exploited to develop the necessary corpus.

3. Growth rate: Wikipedia is an extensive collaborative project within the web, in which articles are published and reviewed by volunteers from around the world. It covers 271 different languages and the Arabic version is ranked 22nd with more than 336K articles[1]. The annual increase in the number of Arabic articles is 25%. This gives an impression of the growth future of this resource in regards to its size and diversity.

4. Unrestricted domain: The articles in the Wikipedia cover a wide range of topical types such as historical, geographical and personal topics, in contrast with newswire corpora. This diversity supports the open domain of knowledge representation.

5.1.2 Arabic Wikipedia and Named Entities

Arabic Wikipedia[2] (AW) is an extensive collaborative project on the web in which articles are published and reviewed by volunteers from around the world. The actual relationship

[1]This statistics is gathered from the official website of Wikipedia for the month of October 2014. URL: http://stats.wikimedia.org/EN/TablesWikipediaAR.htm

[2]http://ar.wikipedia.org

between the NE and AW is that 74% of Wikipedia articles are about NEs (see Section 4.2) This provided the motivation to utilise Wikipedia's underlying structure to produce the target corpus.

To this end, it is beneficial to provide an overview of the critical aspects of the Wikipedia structure:

A. Articles: These can be one of the following:

1. **Normal article:** Each article has a unique title and contains authentic content; i.e. textual data, images, tables, items and links, related to the concept represented in the title. These are in the majority.

2. **Redirected article:** These contain a specific tag to redirect the enquirer to a normal article. For example: for the redirected article titled (بريطانيا العظمى /bryTAnyA AlʕĎm/ 'Great Britain'), there is a redirected tag to (المملكة المتحدة /Almmlkℏ AlmtHdℏ/ 'United Kingdom').

   This tag is written thus #REDIRECTED[[المملكة المتحدة]].

3. **Disambiguation article:** These are used to list all the article titles that share ambiguities.

B. Links types: There are two types of links in Wikipedia and they are described below:

1. **Non-piped links:** this type of links denotes that the display phrase of the link and the article's title are the same. For example: [[London]].

2. **Piped links:** this type of link allows for the text that appears in the contextual data to be different from the actual article it refers to. For example: [[UK|United

Kingdom]], where 'UK' appears in the display text, while 'United Kingdom' refers to the title of the article. Throughout this chapter, the terms 'link' and 'link phrase' are used interchangeably to refer to the same object.

C. Connectivity: Articles in Wikipedia are connected to each other by using the URL links. Exploiting this underlying structure makes the development of the required annotated corpus approachable. For example, a Wikipedia article titled 'London' has links to other articles such as 'United Kingdom' and 'River Thames'.

5.1.3 Compiling the Corpus

The underlying structure of the AW can be exploited to create the annotated corpus by classifying the AW articles into fine-grained NE classes and then mapping this resulting labelling back into the linked phrases in context. To achieve this, our approach works on different steps as follows:

1. Mapping AW[3] into NE taxonomy by classifying the articles into predefined set of NE classes. Therefore, for each AW article, classify the article into the target fine-grained NE class by training an SVM classifier using the training dataset (4000 articles). This step has already been explained in detail in Section 4.7.

2. Preparing the final list of all articles' titles and their tags and then mapping the result of the classification back to the linked phrases in the text.

3. Resolving issues related to the attachment of prefixes and suffixes to the linked phrases. (See Section 5.1.4 for more detail)

4. Developing the Mention Detection Algorithm (MDA) to find successive mentions of NE that have not been associated with links. (See Section 5.1.5 for more detail)

5. Selecting sentences to be included in the final corpus as described in Section 5.1.6.

[3]This project requires taking a snapshot of the dataset dump to work on. I took this snapshot early at the start of this project which was on December 2011.

Figure 5.1 visualises these steps and each of these steps is described in detail in the following sections.

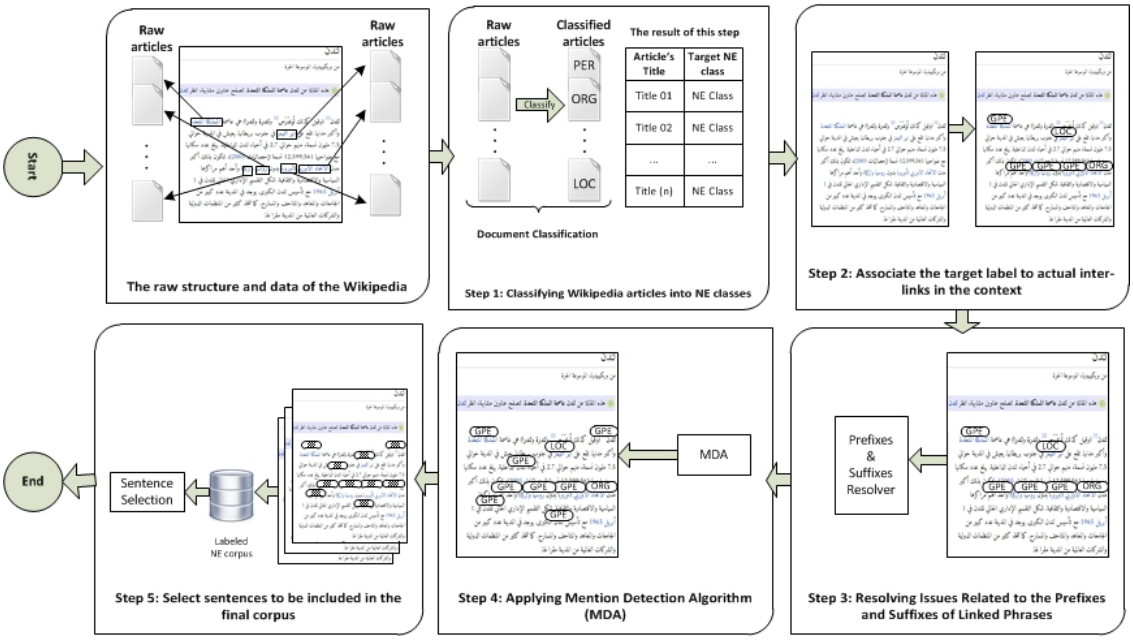

Figure 5.1: Steps taken when automatically developing the fine-grained NE corpus

5.1.4 Prefixes and Suffixes: Issues of Linked Phrases

The way of writing linked phrases in Wikipedia allows for the attaching of text immediately before and after the link. For example, 'a[[b]]c' will be displayed as one word, 'abc', while the link is just associated with 'b'. This issue has a direct relationship to Arabic prefixes and suffixes, a discussion of which will follow.

Prefixes attached: In this case, certain prefixes are attached to the link. For example: (و[[بريطانيا]] /w[[bryTAnyA]]/ 'and[[Britain]]') and (فقام[[فهد]] /fqAm[[fhd]]/ 'stands[[Fahd]]'). After analysing the whole corpus, we found that, this type of prefixes attached to links has two situations:

1. Proclitic attached to the link, such as ((و /w/ 'and'), (ل /l/ 'for'), (ا /A/ 'is'), (ب /b/ 'by') and (ك /k/ 'as')) or a combination of more than one proclitic, such as ((

لل /ll/ ‘for the’), (وال /wAl/ ‘and the’), (ولل /wll/ ‘and for the’) and (وبال /wbAl/ ‘and by the’)).

2. Mistaken attachments: such as (ع[[ثمانية]] /[[ς]]θmAnyħ/ ‘O[[ttoman]]’) where the correct form of writing is ([[عثمانية]] /[[ςθmAnyħ]]/ ‘[[Ottoman]]’).

We analysed the whole text of AW to evaluate the coverage of these links. We found that 98.79% of the proclitics are formed as shown in Table 5.1. The table presents the frequency for each proclitic and the processing method applied to overcome this issue.

Table 5.1: Different cases of prefixes attached to the Wikipedia links

Proclitic	Gloss	Frequency	%	Processing method
و	and	103088	74.5	Space separation
ب	by	11702	8.45	Space separation
ل	for	8151	5.89	Space separation
ال	the	7164	5.17	Merge with the link
لل	for the	2534	1.83	Return لل to its original form ل + ال and then separate ل and merge ال with the link
ك	as	1284	0.92	Space separation
وال	and the	1272	0.91	Separate و and ال and then merge ال with the link
بال	by the	789	0.57	Separate ب and ال and then merge ال with the link
ف	then	332	0.23	Space separation
ا	is	209	0.15	Merge with the link
كال	as the	136	0.09	Separate ك and ال and then merge ال with the link
وب	and by	124	0.08	Space separation
Total		136785	98.79	

Suffixes attached: These are called word-ending links, where the link can be spread to the suffix as well. Analysing the whole textual data of AW shows that the suffixes are:

1. Morphological suffixes attached to the link such as ((ـات /At/ ‘Feminine plural suffix’) - (ها /hA/ ‘Singular feminine pronoun’) and (ين /yn/ ‘Masculine and

feminine dual suffix')). This could result in an ambiguity. For example, cases like ([[مصر]]ي /[[mSr]]y/ '[[Egypt]]ian') will be displayed as مصري, while the actual link between square brackets مصر means Egypt. In this case, Egyptian refers to 'nationality' while Egypt refers to the 'country'.

2. Mistaken attachments: such as ([[محمد]]و[[فهد]] /[[mHmd]]w[[fhd]]/ '[[Mohammed]]and[[Fahd]]') where the spaces before and after the conjunction are missing. Since the suffixes' cases have different variations and long tail of cases, Table 5.2 represents only the most frequent (>= 2%) situations associated with the processing method.

Table 5.2: Different cases of suffixes attached to Wikipedia links

Enclitic	Gloss	Frequency	%	Processing method
ي	1st pronoun	1496	27.86	Merge with the link
ات	Feminine plural suffix	527	9.81	Merge with the link
ا	Dual nominative masculine suffix	439	8.17	Merge with the link
ون	Plural nominative masculine suffix	257	4.78	Space separation
ه	3rd person masculine singular pronoun	131	2.44	Merge with the link
ها	3rd person feminine singular pronoun	113	2.1	Merge with the link
Total		2963	55.16	

5.1.5 Mention Detection Algorithm (MDA)

As a convention, a linking phrase in the text of any Wikipedia article should only be assigned the first time it appears in context; successive mentions of the phrase appear with no link. Therefore, not all NEs are linked every time. Detecting successive mentions works by finding and matching possible NEs in the text that share similarity, to a certain extent, with each phrase in the list of linked NEs. The main goal of this step is to augment the plain text with NE tags and to address some of the lexical and morphological variations that arise when a NE is contextualised.

For example, a NE of (سعود الفيصل /sʕwd AlfySl/ 'Saud Alfaisal') is expected to be repeated in context with either the first name (سعود /sʕwd/ 'Saud') or the last name (الفيصل /AlfySl/ 'Alfaisal') or both together. This can also be difficult when prefixes are used. For example (ولسعود /wlsʕwd/ 'and for Saud'). Therefore, we prepare for and match all the variations of prefixes that can be attached to the NE.

MDA works by importing the list of linked NEs in an article and then dividing them into two groups based on the links' token sizes. A list called 1_{st}Group is used to store all NEs of size one token, and the 2_{nd}Group list includes NEs to sizes greater than one. A preprocessing step is required to expand the NEs in the 2_{nd}Group. The reason for this step is that phrases in this list may not be presented exactly as successive mentions. For example, the NE (محافظة الدرعية /mHAf*Ď*ħ Aldrʕyħ/ 'Diriya Province') is repeated in the context as (الدرعية /Aldrʕyħ/ 'Diriya') without the keyword (محافظة /mHAf*Ď*ħ/ 'Province'). This is also similar to personal NEs, in which some keywords, such as king, president, Mr. and Eng. have been used. Therefore, we prepared two lists of keywords that attached to locational and personal NEs, entitled 'ListOfKeyWords$_{LOC}$' and 'ListOfKeyWords$_{PER}$', respectively (Table 5.3 presented some examples)[4].

Table 5.3: Example list of keywords attached to locational and personal NEs (Full list is presented in Appendix I)

ListOfKeyWords$_{PER}$			ListOfKeyWords$_{LOC}$		
Arabic	Transliteration	Gloss	Arabic	Transliteration	Gloss
الملك	Almlk	King	مدينة	mdynħ	City
الملكة	Almlkħ	Queen	ولاية	wlAyħ	State
الأمير	AlOmyr	Prince	محافظة	mHAf*Ď*ħ	Province
الأميرة	AlOmyrħ	Princess	منطقة	mnTqħ	Region
الوزير	Alwzyr	Minister	بلدة	bldħ	Town
الوزيرة	Alwzyrħ	Secertary	قرية	qryħ	Village
الرئيس	Alr$\widehat{y}$ys	President	حي	Hy	District

Another preprocessing step is related to the writing of personal names. This appears

[4]The full list is presented in the appendix A

clearly in the first token of some Arabic personal names, i.e. (أبو /Obw/ 'father of') and (عبد /ϛbd/ 'slave of'). For example, a NE such as (أبو بكر /Obw bkr/ 'Abo Bakr - father of Bakr') cannot be successively mentioned as (أبو /Obw/ 'Abo') alone, or (بكر /bkr/ 'Bakr'); they should appear next to each other as a single unit of NE. The same situation is applied for personal names starting with (عبد /ϛbd/ 'slave of') like (عبد الله /ϛbdAllh/ 'Abdullah - slave of God').

After performing the preprocessing step with the linked phrases with size > 1, the MDA starts to find and match by going through the text token by token.

In this step, we faced many cases in which morphological variations applied to the NEs in the text, which prevents our algorithm from successfully matching them. For example, an NE such as (محمد /mHmd/ 'Mohammed') can be successively mentioned with certain prefixes, such as (ومحمد /wmHmd/ 'and Mohammed'), (بمحمد /bmHmd/ 'by Mohammed'), (ولمحمد /wlmHmd/ 'and for Mohammed') and (فمحمد /fmHmd/ 'then Mohammed').

After closely inspecting a sample of cases, we noticed that most morphological variations are prefixes attached to the NEs. In very limited cases, such as with (أبو /Obw/ 'father of'), the glyphs must be changed when certain prefixes are attached. For example, when prefixes such as (ك /k/ 'as') and (ب /b/ 'by') or a combination of prefixes, such as (ول /wl/ 'and for'), have been attached to (أبو /Obw/ 'father of'), then the glyph needs to be changed into (أبي /Oby/ 'father of - in genitive cases').

Therefore, we prepared a list of possible prefixes that could be attached to the NEs, as shown in Table 5.4.

Table 5.4: List of possible prefixes attached to NEs

Prefixes attached to NE					
Arabic	Transliteration	Gloss	Arabic	Transliteration	Gloss
ا	A	is	و	w	and
ب	b	by	ل	l	for
ك	k	as	ف	f	then
اف	Af	is then	او	Aw	is and
فب	fb	then by	فل	fl	then for
فك	fk	then as	ول	wl	and for
و ب	wb	and by	وك	wk	and as
افل	Afl	is then for	افك	Afk	is then as
افب	Afb	is then by	اول	Awl	is and for
او ب	Awb	is and by	اوك	Awk	is and as

In the finding and matching step, we injected this list of prefixes into the algorithm to expand the matching process. MDA first finds the exact match without applying any prefixes. If this process fails, then it applies the prefixes and then matches.

Moreover, in certain cases the agglutination characteristic of prefixes with the NEs results in changing the glyphs. For example, if an NE starts with the definitive letters (ال /Al/ 'the'), and attached to prefixes end with the letter (ل /l/ 'for'), such as (ول /wl/ 'and for'), then the resulting glyph is not simply agglutinated. In this case, the prefix (ل /l/ 'for') and the definitive letters (ال /Al/ 'the') should be written as (ولل /wll/ 'and for the'). For example (وللملكة المتحدة /wllmlkħ AlmtHdħ/ 'and for the United Kingdom').

A procedure for handling this issue is presented in Algorithm 1, and the complete MDA algorithm is presented in Algorithm 2.

USAGE: PrefixesBasedExpansion(input NE)
input : ListOfPrefixes = List of prefixes

output: List contains all possible variations of the prefixes attached to the NE_{input}

1 **Define a procedure: PrefixesBasedExpansion**(NE_{input})
2 **foreach** *Prefix* $\in$ ListOfPrefixes **do**
3 **if** NE_{input} *starts with (ال /Al/ 'the') and the Prefix ends with (ل /l/ 'for')* **then**
4 Modifying the NE_{input} by changing (ال /Al/ 'the') into (ل /l/ 'for')
5 $\mathsf{NE}_{prefixed}$ = Concatenate (Prefix & NE_{input})
6 **else**
7 $\mathsf{NE}_{prefixed}$ = Concatenate (Prefix & NE_{input})
8 **end**
9 add $\mathsf{NE}_{prefixed}$ into the output list
10 **end**

Algorithm 1: A procedure for prefixes-based NE expansion

input :

ListOfLinks = List of all Wikipedia links associated with target tag

ListOfKeyWords$_{LOC}$ = List of key words including [country, state, city, town, village, region, district etc.].

ListOfKeyWords$_{PER}$ = List of key words including [king, president, Mr. Eng. Dr. Governor, Minister etc.].

ListOfPersonTags = List of subtypes of person class. Including types like [businessperson, engineer, scientist, athlete etc.].

SpecialTags = List of ['(أبو /Obw/ 'father of')','(عبد /ʕbd/ 'slave of')']

textualData = Actual textual data, one sentence per line

output: List of tokens (one per line) where proper tags have been assigned to each accordingly

Refine ListOfLinks where it should not have links that associated with "O" tag
Sort ListOfLinks based on the number of tokens using Descending order (long to short tokens)
Divide ListOfLinks into two groups:
1_{st}Group = links with token length > 1, hashed by first token
2_{nd}Group = links with token length $= 1$, hashed
foreach *Link* $\in$ 1_{st}Group **do**
    if *tag of Link* $\notin$ ListOfPersonTags **then**
        if *1st token of Link* $\in$ ListOfKeyWords$_{LOC}$ **then**
            tempLink $\leftarrow$ all tokens of Link except the 1st token
            1_{st}Group $\leftarrow$ tempLink (hashed by 1_{st} token)
    else if *1st token of Link* $\in$ SpecialTags **then**
        tempLink $\leftarrow$ 1_{st} token and 2_{nd} token of Link
        add tempLink to 1_{st}Group (hashed by 1_{st} token)
        if *length of Link* $>$ *2* **then**
            tempLink $\leftarrow$ last token
            2_{nd}Group $\leftarrow$ tempLink
        end
    else if *1st token of Link* $\in$ ListOfKeyWords$_{PER}$ **then**
        tempLink $\leftarrow$ all tokens of Link except the 1_{st} token
        add tempLink to 1_{st}Group (hashed by 1_{st} token)
    else
        tempLink $\leftarrow$ 1st token
        2_{nd}Group $\leftarrow$ tempLink
        tempLink $\leftarrow$ last token
        2_{nd}Group $\leftarrow$ tempLink
    end
end
foreach *Token* $\in$ textualData *or PrefixesBasedExpansion(Token)* $\in$ textualData **do**
    if *Token is in* 1_{st}Group **then**
        Match tokens $\in$ 1_{st}Group and assign proper tag
    else
        Match token $\in$ 2_{nd}Group and assign proper tag
    end
end

Algorithm 2: Mention Detection Algorithm (MDA)

5.1.6 Sentence Selection

Our heuristic for selecting the sentences to be involved in the automatically developed corpus was to select only those sentences which had at least one NE. This ensured the creation of a corpus that had the highest possible density of tags. We called this corpus WikiFANE$_{Auto}$

We compiled the corpus for more than 2 million tokens, as shown in Table 5.5. This methodology allows the entire AW to become a tagged, fine-grained NE corpus. Moreover, this version of this dataset is freely available to the research community[5].

Table 5.5: The total number of sentences and tokens for the compiled corpus

Corpus	# of sentences	# of tokens
WikiFANE$_{Auto}$	57126	2,021,177

5.2 Developing Gold-standard Fine-grained Corpora

Since the aim of this work is to conduct a thorough experiment of fine-grained Arabic NEs, we decided to manually create gold-standard, fine-grained NE corpora for the Arabic language, drawing upon two different genres. This will provide a critical benchmark for evaluation and comparison with the automatically constructed corpus.

The first fine-grained corpus is newswire-based, using the same textual data that appears in ANERcorp (Benajiba et al., 2007). The whole corpus was re-annotated to the fine-grained tagset presented in Section 4.1. The second corpus was drawn from the AW. The selection of articles was made using a random heuristic that selected articles discussing a NE, maintaining a fair level of distribution among the classes. Moreover, we restricted the amount of textual data drawn from the Wikipedia articles by avoiding elements such as lists, headings and captions for images and tables.

[5]WikiFANE$_{Auto}$ is freely available at http://www.cs.bham.ac.uk/~fsa081/resources.html

5.2.1 Annotation Strategy and Quality

For both corpora, we applied a similar two-level tagset, presented in Section 4.1 consisting of 8 coarse-grained classes and 50 fine-grained classes. This type of taxonomy suited our need, allowing for a comprehensive evaluation across corpora.

We developed an in-house tool to facilitate the annotation process. Two independent graduate-level native Arabic speakers were asked to annotate the whole corpora. We provided them with extended instructions to guide them in the annotation process, and we conducted several feedback sessions in the early stages of the process to ensure that any difficulties were resolved.

After completing the annotation, we evaluated its quality by calculating the inter-annotation agreement between both annotators. We used the entity F-measure to evaluate the inter-annotation agreement (as in (Hripcsak and Rothschild, 2005; Zhang, 2013)). We called the corpora NewsFANE$_{Gold}$ and WikiFANE$_{Gold}$ referring to newswire-based and Wikipedia-based fine-grained Arabic NE gold-standard corpora, respectively. The details for the gold-standard corpora are listed in the following table.

Table 5.6: Gold-standard corpora and the annotation agreement

Corpus	Size	Genre	Level	Annotation agreement
NewsFANE$_{Gold}$	170K	Newswire	Fine-grained	91%
WikiFANE$_{Gold}$	500K	Wikipedia	Fine-grained	89%

5.3 Corpus-based Evaluation and Comparison

It is important to closely evaluate and compare different developed corpora. The nature of the distribution of NEs is expected to be different to some extent, affecting the performance of learning the probabilistic model. Therefore, we studied the coverage of NEs related to different aspects, including the distribution of length, types and classes.

5.3.1 The Density and Uniqueness of NE

The density represents the coverage of NEs in the level of tokens and phrases. As we can see in Table 5.7, WikiFANE$_{Gold}$ has the greatest density of both levels. This demonstrates that the Wikipedia-based gold corpus tends to represent more NEs in the context than the newswire-based one. Although WikiFANE$_{Gold}$ is 0.7% denser than NewsFANE$_{Gold}$ in the phrase level, it shows a difference (2.4%) in the token level. This indicates that WikiFANE$_{Gold}$ has more variety in the length of NEs than the newswire-based corpus. However, the automatically developed corpus, WikiFANE$_{Auto}$, has a similar density of coverage as the NewsFANE$_{Gold}$.

Table 5.7: The density of NEs on token and phrase levels

Dataset	Token level	Phrase level
NewsFANE$_{Gold}$	10.7	6.7
WikiFANE$_{Gold}$	13.1	7.4
WikiFANE$_{Auto}$	10.8	6.4

Another aspect to consider is the percentage of uniqueness of the NEs for each corpus. In WikiFANE$_{Auto}$, we found that 17% of those NEs are unique, i.e. there are no duplicates (see Table 5.8). This is directly affected by the methodology devised to develop the corpus, in which the total number of the unique NEs has relied upon the total number of the Wikipedia articles talking about the NE. Taking into consideration the annual growth rate of the AW, this approach appears promising for developing an NE corpus in an automatic manner with a reasonably wide coverage of distinct NEs. Although the uniqueness of the NEs in WikiFANE$_{Auto}$ is less in comparison to other gold-standard corpora, this reflects the variety and diversity of the context surrounding the NEs of this corpus. For gold-standard corpora, WikiFANE$_{Gold}$ is denser: 43% of the total NEs are unique, in comparison with NewsFANE$_{Gold}$. This obviously reflects the differences in the nature of those corpora.

Table 5.8: The percentage of uniqueness of the NEs

Corpus	% unique NEs
NewsFANE$_{Gold}$	39
WikiFANE$_{Gold}$	43
WikiFANE$_{Auto}$	17

5.3.2 Lengths of NE Phrases

From Table 5.9, we can see that NewsFANE$_{Gold}$ tends to have more single-word NEs than the other corpora. This is due to differences in the way the NEs are written in a newswire domain. Less than half of the NEs in WikiFANE$_{Auto}$ are single-word, a rate that is slightly higher in WikiFANE$_{Gold}$. The boundaries of multi-word NEs are difficult to detect, especially in Arabic, since the language has a complex morphology and different syntax structure. This is shown in the Wikipedia corpora, i.e. the gold and the automatic.

Table 5.9: The distribution of NEs relative to length.

Corpus	Lengths							
	1	2	3	4	5	6	7	8
NewsFANE$_{Gold}$	58.19	30.77	8	1.73	0.82	0.21	0.2	0.04
WikiFANE$_{Gold}$	51.75	31.55	10.88	3.48	1.34	0.46	0.21	0.12
WikiFANE$_{Auto}$	48.27	37.95	10.22	2.98	0.41	0.11	0.05	0.01

5.3.3 NEs Phrase Structures According to POS

The distribution of the structure of NEs in terms of part of speech (POS[6]) is another important attribute to consider and are summarised in Table 5.10. As we can see, proper nouns 'NNP' in single-word phrases show the importance of performing a POS analysis as a pre-processing step. This is also the case with multi-word NEs in a chain of 'NNPs'. Nevertheless, not all NEs have been parsed as 'NNP', which presents additional challenges. Phrases parsed as common nouns 'NN' have a major ambiguity. Complex phrases such as 'NN NN' further increase the challenge. We found that NewsFANE$_{Gold}$ has 42.72% of single-word proper noun NEs, i.e. NNP. In comparison, both WikiFANE$_{Gold}$ and

[6]AMIRA POS tagger has been used and its tagging accuracy is 96.13% (Diab, 2009)

WikiFANE$_{Auto}$ have lower percentages by: 11.26% and 13.83%, respectively. This distribution explains the syntactic structure difficulties between the newswire- and Wikipedia-based corpora.

Table 5.10: The distribution of the structure of NEs according to the Part of Speech (The POS tagset are presented according to ERTS)

Part of Speech	NewsFANE$_{Gold}$	WikiFANE$_{Gold}$	WikiFANE$_{Auto}$
[NNP]	42.72	31.46	28.89
[NN]	8.66	13.03	12.29
[NNP][NNP]	14.82	10.07	13.37
[NN][NNP]	3.03	5.29	7.15
[NN][NN]	3.51	4.87	5.42
[NN][JJ]	4.6	4.74	5.93
[NNS]	2.82	3.2	1.02
[JJ]	3.53	3.1	4.78
[NNP][NNP][NNP]	2.66	2.7	3.93
[NNP][JJ]	1.33	1.37	1.4
[NN][NN][JJ]	1.42	1.33	1
[NN][NNP][NNP]	0.52	1.14	0.86
[NNS][JJ]	1.07	0.8	0.86
[NNP][NNP][NNP][NNP]	0.37	0.77	0.96
[NN][JJ][JJ]	0.45	0.7	0.53
[NN][NN][NN]	0.44	0.64	0.45
[NN][NN][NNP]	0.18	0.57	0.31
[NN][NNCD]	0.05	0.51	0.06
[NNP][NN]	0.41	0.5	0.76
[NNP][NNCD]	0.05	0.48	0.08

5.3.4 Fine-grained Semantic Class Distribution

This shows the distribution of NEs according to their annotation into fine-grained classes as shown in Table 5.11. In general, the newswire-based corpora tended to include more NEs related to politics, government, commerce, nations and cities, whereas the automatically built corpora scored a high frequency on NE types such as 'nation' and 'population-centre'. Moreover, WikiFANE$_{Gold}$ shows great distribution on most of the fine-grained classes of 'person', 'location', 'facility', 'vehicle' and 'product', compared with other corpora.

Table 5.11: Distribution of fine-grained classes

Class	NewsFANE$_{Gold}$	WikiFANE$_{Gold}$	WikiFANE$_{Auto}$
PER: PERSON	**39.77**	**35.25**	**19.96**
Politician	13.93	9.51	6.94
Athlete	8.21	1.83	1.6
Businessperson	0.5	1.59	0.08
Artist	4.7	3.89	7.04
Scientist	1.18	3.25	1.08
Lawyer	0.17	0.16	0
Police	0.6	1.97	0.38
Religious	1.77	4.06	0
Engineer	0.06	0.39	0.08
Group	3.97	5.43	2.37
Other	4.68	3.17	0.39
ORG: ORGANISATION	**24.95**	**15.85**	**15.05**
Government	3.84	2.83	0.91
Non-Governmental	9.05	2.97	1.31
Commercial	1.75	3.74	1.38
Educational	1.1	1.96	1.15
Media	3.5	1.93	1.21
Religious	0.11	0.31	6.92
Sports	5.08	1.6	2.02
Medical-Science	0.34	0.28	0.09
Entertainment	0.18	0.23	0.06
LOC: LOCATION	**1.2**	**7.34**	**3.96**
Water-Body	0.98	4.35	2.1
Celestial	0.02	1.52	0.61
Land-Region-Natural	0.2	1.47	1.25
GPE: Geo-Political	**27.02**	**24.96**	**55.37**
Continent	0.65	0.99	0.61
Nation	13.79	9.25	17.53
State-or-Province	1.5	3	2.64
County-or-District	0.42	0.54	0.68
Population-Center	9.32	10.17	31.52
GPE-Cluster	1.34	1.01	1.5
Special	0	0	0.89
FAC: FACILITY	**3.52**	**5.79**	**1.76**
Building-Grounds	2.93	3.46	1.43
Subarea-Facility	0.17	0.46	0.02
Path	0.21	1.19	0.21
Airport	0.2	0.55	0.1
Plant	0.01	0.13	0
VEH: VEHICLE	**0.98**	**3.27**	**0.22**
Land	0.76	0.17	0.08
Air	0.2	2.39	0.09
Water	0.02	0.71	0.05
WEA: WEAPON	**0.05**	**0.69**	**0.44**
Blunt	0	0.07	0.01
Exploding	0	0.1	0.09
Sharp	0	0.01	0.21
Chemical	0	0.01	0
Shooting	0.02	0.09	0.06
Projectile	0.03	0.31	0.03
Nuclear	0	0.1	0.04
PRO: PRODUCT	**2.52**	**6.84**	**3.26**
Book	1.28	1.78	0.81
Movie	1.04	0.7	1.23
Sound	0	0.52	0.17
Hardware	0.07	1.44	0.34
Software	0.1	2.03	0.35
Food	0.02	0.22	0.35
Drug	0.01	0.15	0.01

5.3.5 Average Sentence Length

The average number of tokens per sentence is another way to compare the characteristics of different corpora. We found that the length of the sentences increases by (1% to 4%) when one or more NEs were involved and depends on the corpus. This is clearly shown in NewsFANE$_{Gold}$ and WikiFANE$_{Gold}$ whereas WikiFANE$_{Auto}$ shows no variation. This might be due to less diversity in the NEs, as we already mentioned in Section 5.3.1. The variety between WikiFANE$_{Gold}$ and WikiFANE$_{Auto}$ suggests that Wikipedia, as a public resource of knowledge, is diverse in terms of domains and genres, including historical, personal, geographical and scientific topics, in comparison to the newswire-based corpus.

Table 5.12: Average sentence length in the terms of the number of tokens

Dataset	All Sentences	Sentences including $\geqslant$ one NE
NewsFANE$_{Gold}$	35	39
WikiFANE$_{Gold}$	31	34
WikiFANE$_{Auto}$	38	38

5.4 Chapter Summary

In this chapter, we developed fine-grained Arabic NE corpora using two different approaches. The first approach, as explained in Section 5.1 was to construct a scalable corpus in an automatic manner by exploiting the richness of the AW. This approach involved recruiting document classifications and implementing an MDA to tag successive mentions in the context. Using this methodology, we produced constantly evolving NE resources that could exploit the yearly growth rate of the AW. Moreover, we developed two gold-standard corpora from different genera, i.e. newswire- and Wikipedia-based, as seen in Section 5.2. In Section 5.3, we conducted a series of corpus-based comparisons and evaluations in order to demonstrate the differences between corpora in some aspects, with respect to the task of NER. In the following chapter, we will discuss the learning of a supervised ML probabilistic model in order to develop a baseline model for fine-grained

Arabic NEs.

Part IV

FINE-GRAINED NAMED ENTITY RECOGNITION

This part discusses different contributions towards the development of the fine-grained NER for Arabic. This part starts with Chapter 6 where the methodology of learning different ML classifiers is presented by relying on window-based local feature representation. Chapter 7 discusses the advised approach to represent features by relying on dependency structure instead of a window-based method to exploit sentence level knowledge. In Chapter 8, further contribution is presented where the richness of raw textual data is exploited to extract useful knowledge by using clustering techniques.

Chapter 6

Fine-grained Named Entity Recogniser

Chapter Synopsis

In the previous chapter we presented our methodology to build fine-grained NE corpora from different sources and by using different approaches (i.e. automatically and manually). In this chapter, we will present the implementation of the fine-grained NER. Section 6.1 will demonstrate the pipeline architecture of NER that relies on different components. In Section 6.2, we will learn a Maximum Entropy (ME) classifier to develop the baseline model for our NER. After that, learning different classifier such as Conditional Random Fields (CRF) will be discussed in section 6.3. Section 6.4 will discuss the effect of injecting external knowledge in the probabilistic knowledge. In the following section, we will demonstrate different ways of encoding the NEs in the classification process. In section 6.6, we conduct an in-depth evaluation to analyse the misclassified results.

6.1 The Pipeline Architecture of NER

Our implementation of a fine-grained NER system is based on a pipeline architecture which consists of several steps. The output of one step is the input of the next one. In this section we will discuss the role of each step.

The following figure illustrates the components of the pipeline architecture of NER.

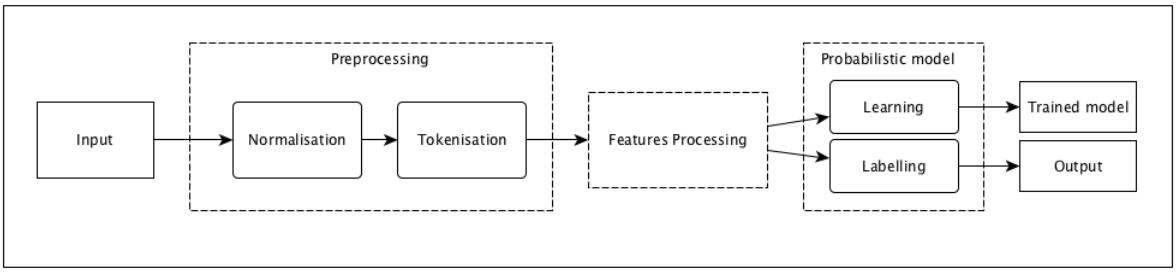

Figure 6.1: The pipeline architecture of NER

6.1.1 Preprocessing

Since we deal with Arabic text in its raw form, we have to design and perform a preprocessing step which facilitates dealing with the textual data in the later steps. These are presented as follows:

1. Normalisation: One of the important preprocessing steps is normalisation. The aim of this step is to ensure that the textual data is presented in their correct form for processing. Although, the alphabet of Arabic consists of 28 letters there are additional shapes used to represent some variations in the context. Those are presented below.

1. The letter (ا /A/) can be presented with different variations such as (آ, إ, أ, ا /A, O, I, $\bar{A}$/) depending on their use in the context. However, there are less strict rules to use which, as a result, introducing ambiguity. For example (أحمد /OHmd/ 'Ahmed') and (احمد /AHmd/ 'Ahmed')

2. The letter (ي /y/) can be mistakenly written as (ى /ý/) where they have very similar glyphs while they are different letters. This is also the same with (ة /ħ/)

and (ه /h/).

3. The variation of writing (ء /'/ the letter Hamza) where it can be written in isolation such as (ء /'/) or accompanied by letters as in (ؤ /W/) and (ئ /$\widehat{y}$/).

4. The Kashida (ـ) is used to elongate the appearance of letters such as (فـــهـــد /f_h_d/ Fahd).

5. Diacritics where they are optionally used in the context. (See Table 2 for all diacritics in Arabic.)

To avoid sparsity, we decided to normalise some orthographical shapes of some similar letters that have no effects when NER is concern. Therefore, different shapes of letter (آ, إ, أ, ا /A, O, I, $\bar{A}$/) have been automatically changed into plain (ا /A/). This step helps to overcome the spelling mistake in the context. Some researchers, such as Xu et al. (2002), suggest normalising the letter (ي /y/) but we noticed that normalising this letter will increase the ambiguity to detect NEs such as (علي /ʕly/ 'Ali') comparing with the proposition (على /ʕlý/ 'on'). We also removed diacritics and unified the encoding of digits, punctuations and symbols.

2. Tokenisation: Tokenisation is another important preprocessing step. By tokenisation, we mean to separate prefixes and suffixes from the stem of the word. This step aims to reduce sparsity. For example, prefixes attached to the front of the tokens increase the sparsity. Therefore, a classifier which sees a NE such as (وخالد /wxAld/ 'and Khalid') that starts with a conjunction prefix and is written as one token, and then is asked to

classify the token (خالد /xAld/ 'Khalid') with no prefixes will not be able to correctly classify it.

In order to resolve this issue, we tokenised the text by relying on AMIRA which was developed by Diab (2009). We closely inspect the cases where prefixes and suffixes are attached to NEs. We decided to use the scheme (conj+prep+suff)[1] which means separating only conjunctions, prepositions and suffixes. This scheme does not separate the definite article (ال /Al/ 'the') where it is considered as part of NE such as in personal family names such as (العتيبي /Alʕtyby/ 'Alotaibi').

6.1.2 Feature Processing

The performance of the probabilistic model, i.e. learning algorithm, has been heavily affected by the involved set of features. The probabilistic model produces a higher performance as the features become more informative. In the context of NER, several set of features have proven to benefit the classification process. They are traditionally extracted to represent orthographical, contextual, morphological and syntactical features. More details of the selected features in the baseline model will be presented in Section 6.2.2. One of the important contributions of this thesis is to recommend a novel set of features that overcome the limitation of traditional ones and these will be discussed in Chapter 7 and 8.

6.1.3 Probabilistic Model

The core component of any supervised ML approach is the probabilistic model where it learns from training data and then tries to predict on unseen text. Therefore, this model works in two phases; training and then labelling (i.e. testing). The probabilistic model

[1]In this scheme the conjunctions, prepositions and suffixes are separated by white space.

we rely on is based on sequence labelling as mentioned in Section 3.3.2.1. The input to this model is one token per line associated with the extracted features in columns. The last column is the target class. Table 6.1 shows an example of snippet of text after doing the preprocessing and the feature processing steps. In this example, for each token there are two extracted features, i.e. part of speech (POS) and the base phrase chunk (BPC). The probabilistic model used the provided set of feature to learn and then to predict the target tag.

Table 6.1: An example of the pre-processed input text

Token	Gloss	POS	BPC	Tag
اعتذر	apologies	VBD	B-VP	O
الرئيس	president	NN	B-NP	O
السابق	former	JJ	I-NP	O
ل	for	IN	B-PP	O
مصر	Egypt	NNP	I-PP	B-Nation
و	and	CC	I-PP	O
ايران	Iran	NNP	I-PP	B-Nation
-	-	PUNC	I-PP	O
خلال	through	NN	I-PP	O

6.2 Baseline Model based on Maximum Entropy (ME)

Since there is no comparative work in the form of a fine-grained Arabic NER systems, we developed a baseline model based on Maximum Entropy (ME)[2] to serve as benchmark for forthcoming evaluations. The reason for selecting this probabilistic model was that it requires less training time in comparison with other classifiers, such as Support Vector Machine (SVM), and it has been used as a baseline model for similar tasks, i.e. the POS tagger (Toutanova and Manning, 2000).

[2]For ME as state-of-the-art of the baseline model, we relied on Wapiti - A simple and fast discriminative sequence labelling toolkit (Lavergne et al., 2010)

6.2.1 Dataset

The common approach in the literature to divide the dataset is by having two portions, i.e. training and test. This approach could suffer from the over-fitting problem where the classifier tries to minimize the classification error by utilising the provided features and classification parameters. To avoid this issue, it is advised to divide the dataset into three parts (i.e. training development and test) and each subset has to be chosen randomly. In this way, the development (i.e. validation) set is introduced which is used to tune the features and the training parameters. This makes sure that the classifier is not over-fit on the training portion, but it take into consideration the development set as well. It is important to emphasise that the series of experiments was conducted in a cumulative manner, in which the latter one was built on the top of the previous one or one that had been otherwise mentioned. Moreover, due to the limitations of computation power and the space allocated for our machine specification, we only selected a portion of WikiFANE$_{Auto}$ with a size of 500K tokens. Table 6.2 shows each corpus and its size.

Table 6.2: The size of the training, development and test for each corpus

Corpus	Type	Training	Dev	Test
NewsFANE$_{Gold}$	gold-standard	120K	25K	25K
WikiFANE$_{Gold}$	gold-standard	350K	75K	75K
WikiFANE$_{Auto}$	automatically-developed	354K	73K	73K

6.2.2 Features Extraction

For the baseline model, we extracted the features by following the best practise in literature as reviewed in Chapter 3. The features used in this study are divided into three groups as follows:

6.2.2.1 Lexical Features

This set aims to capture the important surface features in the lexical and contextual levels. This involves:

1. The current token T_i.

2. The window of two tokens in both sides T_{i-2} ; T_{i-1} ; T_i ; T_{i+1} ; T_{i+2}.

3. Character level features: Suppose a token consists of a sequence of characters whereby C_1 and C_n represent the first and last character respectively, therefore the following features are used (C_1); (C_1 + C_2); (C_1 + C_2 + C_3); (C_{n-2} + C_{n-1} + C_n); (C_{n-1} + C_n); (C_n). This set of features aims to capture important knowledge in the presence of affixes that remained after doing the tokenisation such as the definite article (ال /Al/ 'the').

6.2.2.2 Morphological Features

For morphological features, we exploited gender, number and person as they are provided out of AMIRA toolkit. Moreover, the stem of a token is also prepared in order to present the token in its lighter form by removing any prefixes and suffixes. We relied on the algorithm provided by Taghva et al. (2005) to extract the stems.

6.2.2.3 Shallow Syntactical Features

Parts of speech (POS) have been proven to be a very helpful feature in NER in different languages such as English (Florian et al., 2003). Thus, AMIRA is also used to extract the target POS for a token. A window like POS[T_{i-2}], POS[T_{i-1}] , POS[T_i] , POS[T_{i+1}] , POS[T_{i+2}] is generated.

We also extracted the representation of the base phrase chunk (BPC). In this representation, tokens are grouped in phrase such as Noun and Verb Phrases, i.e. NP and VP, and so on. This is useful, because NEs are expected to span over NPs.

6.2.3 A Pilot Experiment for Baseline Model

For the baseline model, we used a traditional scheme representation, i.e. BIO, of NEs. In the BIO scheme, the first token in the NE is tagged by 'B-XXX'[3] and successive tokens use 'I-XXX'. Where there is no NE token, 'O' is used. After learning the ME by using the mentioned set of features, the results on the development and the test dataset are presented in Table 6.3.

Table 6.3: The results of the baseline model based on the ME classifier

Corpus	Development			Test		
	P	R	F	P	R	F
NewsFANE$_{Gold}$	63.46	52.25	57.31	55.77	45.26	49.97
WikiFANE$_{Gold}$	41.72	37.14	39.3	46.17	36.51	40.77
WikiFANE$_{Auto}$	63.19	37.93	47.4	70.68	38.79	50.09

ME performs the best over WikiFANE$_{Auto}$ where the precision metric scores the highest. On the other hand, WikiFANE$_{Gold}$ yields the lowest performance. This result shows that the gold-standard Wikipedia-based corpus is more difficult than the automatic one. This is due to the density and uniqueness of WikiFANE$_{Gold}$ comparing to WikiFANE$_{Auto}$ as discussed in Section 5.3.1. NewsFANE$_{Gold}$ performs well over the development portion of the data whereas both precision and recall show degradation that affects the F-measure over the test data. Interestingly, the precision metric for all corpora has superiority over the recall that infers the overall difficulty is of retrieving NEs.

6.3 Using Conditional Random Fields (CRF) as a Different Classifier

The efficiency of the NER can be affected by a variety of different issues (Ratinov and Roth, 2009). Among these is the choice of a suitable probabilistic model. Support Vector Machine (SVM) and Conditional Random Fields (CRF) performed better than ME for use with POS tagging (Avinesh and Karthik, 2007). In this experiment, we use the

[3]XXX is replaced by the type of NE such as PER for person and so on

same scheme and features to train another classifier, i.e. the CRF probabilistic model. The results presented in Table 6.4 show improvement across all corpora. WikiFANE$_{Gold}$ and NewsFANE$_{Gold}$ returned the highest improvement respectively. This shows that the selected probabilistic model can utilise selected features to perform better predictions than the ME. The recall metric has boosted across all corpora resulting in retrieving more NEs compared with the ME model. Learning CRF model over the newswire corpus shows the highest score of F-measure comparing with Wikipedi-based corpora.

Table 6.4: The results of the CRF classifier using the same features as used in the baseline model. (+|− represents the variation compared with the previous experiment)

Corpus	Development			Test			+|−
	P	R	F	P	R	F	
NewsFANE$_{Gold}$	76.52	54.91	63.94	69.94	48.91	57.56	**7.59**
WikiFANE$_{Gold}$	59.35	42.09	49.25	63.54	39.94	49.05	**8.28**
WikiFANE$_{Auto}$	78.26	42.26	54.88	80.72	41.11	54.48	**4.39**

6.4 Applying External Knowledge

The intention when relying on external knowledge, i.e. the gazetteer, was to increase the accuracy of the model. In this experiment we used WikiFANE$_{Gazet}$, the fine-grained gazetteer developed and presented in Chapter 4, to enrich the features set, applying this knowledge. Table 6.5 shows the results for each corpus and the level of improvement once the gazetteer has been used.

Table 6.5: The results of the CRF classifier with the gazetteer used for external knowledge

Corpus	Development			Test			+|−
	P	R	F	P	R	F	
NewsFANE$_{Gold}$	81.1	59.53	68.66	72.54	53.1	61.32	**3.76**
WikiFANE$_{Gold}$	66.42	40.62	50.41	71.58	40.33	51.59	**2.54**
WikiFANE$_{Auto}$	85.6	64.4	73.5	87.45	57.68	69.51	**15.03**

The results show improvements across all corpora. As we can see, WikiFANE$_{Auto}$ scores the highest improvement. Although, WikiFANE$_{Gold}$ corpus and WikiFANE$_{Gazet}$

gazetteer share similar genre, i.e. from the Arabic Wikipedia, WikiFANE$_{Gold}$ scores the lowest improvement. This is because that the coverage of the unique NE in this corpus is high compared with other corpora as seen in Section 5.3.1. Meaning, there is a high number of manually annotated NEs that have no matches in the gazetteer list.

6.5 Encoding Scheme

Current approaches at the coarse-grained level use only the BIO scheme, such as (Benajiba et al., 2010; Abdul-Hamid and Darwish, 2010). In fact, this scheme is not the only way to encode NEs and the selection of such a scheme should be made carefully. Thus, we will investigate both the IO and BILOU schemes. The former merely assigns 'I-XXX' to NE for all tokens with no differentiation in the position of the token. The latter encoding scheme uses 'B-XXX', 'I-XXX', 'L-XXX' for the first, internal and final tokens of the NE respectively. Single token NE is encoded using 'U-XXX'.

To illustrate those schemes, Table 6.6 shows two NEs, i.e. (لندن /lndn/ 'London') and (الولايات المتحدة الأمريكية /AlwlAyAt AlmtHdħ AlOmrykyħ/ 'United States of America'), which have different sizes of tokens encoded with different schemes accordingly.

Table 6.6: Two examples of different encoding schemes

	Tokens	Gloss	IO	BIO	BILOU
First example	لندن	London	I-LOC	B-LOC	U-LOC
Second example	الولايات	States	I-LOC	B-LOC	B-LOC
	المتحدة	United	I-LOC	I-LOC	I-LOC
	الأمريكية	America	I-LOC	I-LOC	L-LOC

We evaluate different encoding schemes by learning a classifier with a similar configuration for each scheme. As shown in Table 6.7, the current scheme used in the literature, i.e. BIO was not the proper selection as it scored the lowest for most cases. Both schemes, i.e. IO and BILUO, show improvements for different corpora. One of the drawbacks of IO representation is that it fails to properly capture the actual boundary of NE especially

between adjacent NEs. Therefore, we decided from this point onwards to use the scheme BILUO for upcoming experiments, instead of BIO.

Table 6.7: The performance of different encoding schemes (The bold style is used for the highest F-measure score)

Corpus	IO			BIO			BILOU		
	P	R	F	P	R	F	P	R	F
NewsFANE$_{Gold}$	74.16	50.48	60.07	72.54	53.1	61.32	73.07	53.34	**61.67**
WikiFANE$_{Gold}$	63.62	45.95	53.36	71.58	40.33	51.59	68.13	44.78	**54.04**
WikiFANE$_{Auto}$	90.22	58.3	70.83	87.45	57.68	69.51	88.69	60.37	**71.84**

6.6 Error Analysis

The precision, recall and F-measure are formal metrics to evaluate how effective the classifier in tagging. In this section, we applied other evaluation methods to analyse errors in different interpretations. We use similar method of error analysis, presented in this section, in order to evaluate different approaches presented in Chapters 7 and 8.

6.6.1 Confusion Matrix

A confusion matrix is a way to evaluate and present the differences between the predicted and correct tagging. It shows each class and how many times each class has been correctly or mistakenly predicted. We decided to present the confusion matrix on the coarse-grained classes for the sake of elegant presentation. Table 6.8, 6.9 and 6.10 show the confusion matrix of NewsFANE$_{Gold}$, WikiFANE$_{Gold}$ and WikiFANE$_{Auto}$ respectively. The results presented in bold style in the diagonal show the correct tagging. For example, in Table 6.8, 'PER' class has been correctly predicted 804 times whereas 2 times been mistakenly tagged to 'ORG'. In the same example, there are 330 NEs were not be able to be predicted and thus been given 'O' tag instead.

Table 6.8: Confusion matrix of NewsFANE$_{Gold}$

	PER	ORG	LOC	GPE	FAC	VEH	WEA	PRO	O
PER	**804**	2	0	0	1	0	0	0	330
ORG	6	**415**	0	81	0	1	0	0	388
LOC	0	0	**34**	3	0	0	0	0	18
GPE	2	2	0	**581**	0	0	0	1	159
FAC	4	33	0	4	**20**	0	0	0	38
VEH	0	0	0	0	0	**8**	0	0	33
WEA	0	0	0	0	0	0	**0**	0	0
PRO	2	2	0	3	1	0	0	**29**	102
O	29	38	0	17	2	1	0	1	**24752**

In WikiFANE$_{Gold}$ as shown in Table 6.9, the classifier shows notable confusion of tagging NEs. For example, 41 of 'ORG' NEs have been tagged as 'GPE' places. On the other hand, there are 145 'GPE' NEs have been assigned 'LOC'. The classifier could not detect any of the 'WEA' entities where all are assigned 'O'. Moreover, the total number of undetected entities (and therefore assigned 'O') is 2998. This clearly shows that, the classifier could not easily able to differentiate between certain classes to predict the correct tags.

Table 6.9: Confusion matrix of WikiFANE$_{Gold}$

	PER	ORG	LOC	GPE	FAC	VEH	WEA	PRO	O
PER	**2831**	7	5	31	2	0	0	4	1342
ORG	27	**994**	2	41	21	1	0	3	465
LOC	5	0	**258**	33	6	0	0	0	110
GPE	23	5	145	**1917**	2	0	0	2	776
FAC	4	0	6	2	**253**	2	0	0	102
VEH	0	0	0	0	0	**141**	0	0	59
WEA	0	0	0	0	0	0	**0**	0	6
PRO	3	0	2	2	2	2	0	**68**	138
O	176	69	38	127	25	9	0	17	**67527**

Table 6.10 shows that confusion matrix of the result of tagging over the WikiFANE$_{Auto}$ corpus. As we can see, the classifier has less confusion in this corpus comparing with WikiFANE$_{Gold}$. In this corpus, the classifier fails to detect any of the 'VEH' NEs.

Table 6.10: Confusion matrix of WikiFANE$_{Auto}$

	PER	ORG	LOC	GPE	FAC	VEH	WEA	PRO	O
PER	**1053**	2	1	16	2	0	0	0	594
ORG	2	**289**	0	13	0	0	0	0	215
LOC	4	0	**419**	5	0	0	0	0	199
GPE	3	3	8	**3489**	0	0	0	0	1585
FAC	0	0	1	3	**106**	0	0	0	101
VEH	0	0	0	0	0	**0**	0	0	9
WEA	0	0	0	0	0	0	**5**	0	24
PRO	0	1	0	1	0	0	0	**25**	155
O	115	10	16	201	8	0	0	2	**67935**

6.6.2 NEs Phrase Length

Another important aspect to evaluate and to analyse is where the classifier faced difficulties in terms of the length of the NEs. We expected that multi-words NEs are more difficult to predict compared with single-word NEs. For each corpus, we fetched all NEs and divided them into two groups based on their length, the first and the second groups are for single- and multi-words NEs respectively. For each group, we calculate the number and percentage of mistakenly tagged NEs that the classifier fails to predict.

As we can see in Table 6.11, the classifier has 39% error rate of prediction for single-words NEs over NewsFANE$_{Gold}$. In multi-words, the classifier performs worse and faced difficulties as it scores 53% of mistakenly predicted NEs. This shows that the classifier could not able to delimit the boundary of more than half of the multi-words NEs in proper manner.

For WikiFANE$_{Gold}$, the classifier has 46% and 45% error rates of tagging single- and multi-words NEs respectively. A reason behind this difficulty is the increased level of the density and the uniqueness of the NEs in this corpus as we discussed this in Section 5.3.1.

The result of the analysis is different for WikiFANE$_{Auto}$. The classifiers struggled to correctly classify single- compared to multi-words NEs. Among single-words NEs, there are 43% have not been correctly predicted whereas the classifier has performed better on multi-words NEs comparing with other corpora.

Table 6.11: Error analysis of length of NE phrases

Group	NewsFANE$_{Gold}$		WikiFANE$_{Gold}$		WikiFANE$_{Auto}$	
	#	%	#	%	#	%
Single-word:	376	39	1462	46	1449	43
Multi-word:	444	53	1085	45	670	32

6.6.3 Fine-Grained Classes of the Same Parent

The confusion matrix applied to Section 6.6.1 was presented at coarse-grained level due to space limitation and elegant presentation. In this section, another way to analyse the errors at the fine-grained level is given especially for cases where fine-grained classes share the same parental coarse-grained type.

The classification of fine-grained classes that share the same coarse-grained parent class is not an easy task. For example, the classification of 'politician' and 'nation' is easier than the classification of 'athlete', because 'politician' and 'athlete' share the same parent: 'person'. The reason behind this difficulty is that the fine-grained classes with the same parents tend to share similar contexts, which makes it harder to capture informative clues and evidences.

In this analysis, we reported the total number of times that NEs were mistakenly classified at the fine-grained level where they share the same parent. For example, if an NE is assigned to 'athlete' instead of 'politician', we consider the classifier to have struggled to predict the correct fine-grained class, because both classes share the same parent.

For NewsFANE$_{Gold}$, as can be seen in Table 6.12 , there are 136 NEs (14.09% of which are misclassified NEs) that involve misclassification at the fine-grained level and share the same coarse-grained parental class. In WikiFANE$_{Gold}$, the classifier has struggled less where it has failed to predict 248 NEs (8.07%). In WikiFANE$_{Auto}$, the performance of the classifier was better than in WikiFANE$_{Gold}$ and NewsFANE$_{Gold}$. There are only 33 NEs (0.99%) that have not been correctly classified at the fine-grained level of the shared or similar parent.

Table 6.12: Error analysis of tagging fine-grained NEs that share same parent

	NEWSFANE$_{Gold}$	WIKIFANE$_{Gold}$	WIKIFANE$_{Auto}$
#	136	248	33
%	14.09	8.07	0.99

6.7 Chapter Summary

In this chapter, we demonstrated the development of the baseline model of the fine-grained NER for Arabic. We relied on the pipeline architecture to develop the model as seen in Section 6.1. In section 6.2, the development of the baseline model by learning a classifier based on ME probabilistic model was discussed. This followed by learning different probabilistic model, i.e. CRF. In section 6.4, we showed the importance and the effect of injecting external knowledge, i.e. gazetteer, in the classification process. After that, we carefully evaluate different encoding scheme where we notice differences for each one. This chapter ends by analysing the error of the classification process (see Section 6.6). Thus far, we representing the features in the classification process as a window-based of local features. In the following chapter, we will present a new approach of representing the features by relaying on the dependency structure to overcome the drawbacks of the traditional window-based one.

Chapter 7

Dependency-based Approach to Fine-grained NER

Chapter Synopsis

In the previous chapter we discussed the development of fine-grained NER using different probabilistic models with different properties. In this chapter, we present a new approach to representing the features by relying on the dependency structure of the Arabic sentence. With regarded to NER, the dependency structure has been favoured over the constituent structure. This is because that the dependency structure provides a unique way to represent the connections between words according to their relationship. Moreover, those connections are labelled according to their roles such as subject and object. Section 7.1 starts by discussing the limitations of the traditional window-based representation. In Section 7.2 the actual dependency representation is presented in detail. Section 7.3 presents the hybrid approach, which combines both the window- and dependency-based representations in one model. This chapter ends with an analysis of errors according to the procedure specified in the previous chapter (see Section 6.6).

7.1 The Limitations of Window-based Representation

The current representation of the sequence tagging classifier discussed in Chapter 6 involves using a predefined window of tokens (e.g. with size 5, including the current token) to capture local evidence. We call this a window-based representation. We used this representation for all experiments presented in Chapter 6. This representation has some limitations, namely that:

1. It is restricted to capturing only local evidence. Since the window-based representation predefines the size of tokens, this will limit the evidence captured and is limited to the clues of adjacent features only. Expanding this window further has proven to have a negative effect on the overall performance of the classifier as shown by Benajiba et al. (2009b).

2. It fails to capture the relationship between dependent tokens, particularly for long sentences, and multiword NEs.

3. Because Arabic has a relatively free word order, the window-based feature representation cannot adequately capture the order variation of sentence structures.

In this chapter, we investigate a new approach in which to capture clues that go beyond the size of the window by relying on the dependency structure of the Arabic sentences.

7.2 Dependency-based Representation

In this thesis, a new approach has been devised to utilise further evidence within a sentence to support the classification process. The key idea informing this approach is reliance on the dependency-based representation of sentences to extract valuable features.

The dependency structure represents syntax, where a sentence is analysed by connecting its words in a word-to-word relationship. These relationships specify the head and

dependent tokens contextually and assign a grammatical role for each token, e.g. subject, object and modifier.

To elaborate on the amount of knowledge that can be utilised based on the dependency structure, consider the following sentences:

- (‘قال رئيس مجلس اتحاد المحاكم الاسلامية في الصومال شيخ شريف شيخ أحمد في ...الخ’ /qAl r$\widehat{y}$ys mjls AtHAd AlmHAkm AlAslAmy$\hbar$ fy AlSwmAl $\check{s}$yx $\check{s}$ryf $\check{s}$yx OHmd fy ...Alx/ ‘The head of the Council of the Islamic Courts Union, Sheikh Sharif Sheikh Ahmed, said in Somalia, in...etc.’)

- (‘يقول شارلز مورفي السياسي الانجليزي بعد الزيارة الأخيرة التي قام بها جون ميجور رئيس وزراء بريطانيا ...الخ’ /yqwl $\check{s}$Arlz mwrfy AlsyAsy AlAnjlyzy bςd AlzyAr$\hbar$ AlOxyr$\hbar$ Alty qAm bhA jwn myjwr r$\widehat{y}$ys wzrA’ bryTAnyA ...Alx/ ‘Charles Murphy, the English politician, said after the recent visit by John Major, Britain’s prime minister ... etc.’)

- (يذكر أن صلاد حسن انتخب رئيساً للصومال في اغسطس آب ٢٠٠٠ /y$\eth$kr On SlAd Hsn Antxb r$\widehat{y}$ysA$\tilde{a}$ llSwmAl fy AγsTs $\overline{A}$b 2000/ ‘It was mentioned that Salad Hasan was elected as president of Somalia in August 2000’)

The dependency representation and an English gloss for each example are shown in Figures 7.1, 7.2 and 7.3[1]. The parsed output includes a new set of information that can be utilised and features as follows:

1. **Head and Dependent Relationship:** The relationship between the head and the dependent is one of the most important features to capture. Consider the token (شيخ /$\check{s}$yx/ ‘Shaikh’), in Figure 7.1; the head (رئيس /r$\widehat{y}$ys/ ‘the head of’) is located far away and cannot be captured in the local window-based representation. Moreover, the vice versa relationship between the dependent and head is also useful. Consider the example in Figure 7.2: the token (جون /jwn/ ‘John’) has two dependents (

[1]Since all examples are parsed by using CATiB dependency parser, the POS tag set is different from RTS shown in Table 3. CATiB dependency parser extends the basic CATiB POS tag set from 6 into 44 different tags (Habash, 2010, p.83).

ميجور /myjwr/ ‘Major’) and (رئيس /rŷys/ ‘Prime’) where the latter dependent (i.e. ‘رئيس’) gives a useful clue of the way in which it has been used in political contexts. The sequence of heads or dependents can also be utilised in the same way.

2. **Sibling Relationship:** The sibling tokens are those dependent on the same head. Siblings can be located near each other in context, or appear at a distance. For example: the sibling of the token (شيخ /šyx/ ‘Shaikh’) that is (مجلس /mjls/ ‘council’), in Figure 7.1, is expected to appear in a political context, which gives a clue to the target NE class. Meanwhile, the token (في /fy/ ‘in’) is also a sibling, but can be ignored, as it is a stop word. This is also the case in the example presented in Figure 7.3, where the token (صلاد /SlAd/ ‘Salad’) is a sibling of the token (انتخب /Antxb/ ‘elected’), which relates to the political context.

3. **Syntactic Roles:** The syntactical roles also benefit by being utilised to capture NEs in context. Among those with a concern for NER are:

   - **SBJ and OBJ:** defines which subject and object roles are assigned to the head token of the NE. For example, the tokens (صلاد /SlAd/ ‘Salad’) and (شارلز /šArlz/ ‘Charles’) are tagged as subjects.
   - **Idafa:** the Idafa chain is another important syntactical role, which helps to identify multiword NEs. The Idafa construction denotes a combination of two nouns, where the first and second are called possessor and possessed respectively. The possessor and possessed have construct and genitive cases respectively. This is equivalent to compound nouns in English, such as ‘Noun1 Noun2’, ‘Noun1 of Noun2’ and ‘Noun2’s Noun1’. For example: the token (مورفي /mwrfy/ ‘Murphy’) is tagged as an Idafa of its previous token (شارلز /šArlz/ ‘Charles’), where this indicates a multiword NE. This is also the case

for the example (مجلس اتحاد المحاكم الإسلامية /mjls AtHAd AlmHAkm AlIslAmyħ / 'Council of the Islamic Courts Union') where all tokens are assigned an Idafa role except the last token.

- **Flat Relationship (−):** is a special role undertaken by a CATiB dependency parser for the sequence of proper nouns. For example: NEs such as (شيخ شريف شيخ أحمد /šyx šryf šyx OHmd/ 'Sheikh Sharif Sheikh Ahmed'), in which all tokens are assigned a flat relation.

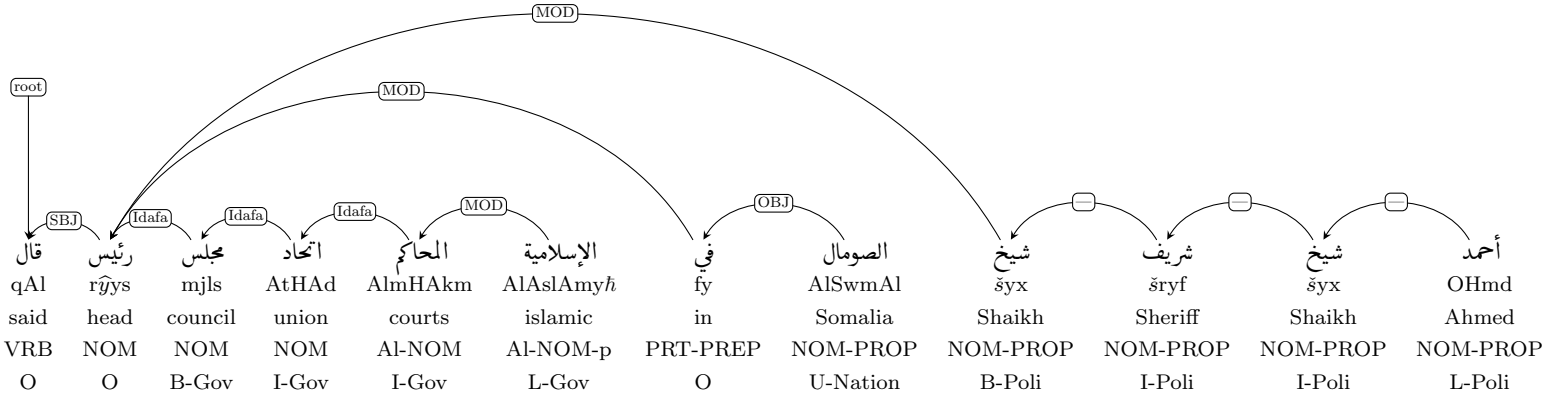

Figure 7.1: The first example of a dependency structure. The rows show the Arabic token, Buckwalter transliteration, English gloss, POS and NE tag, respectively (the sentence is displayed left to right).

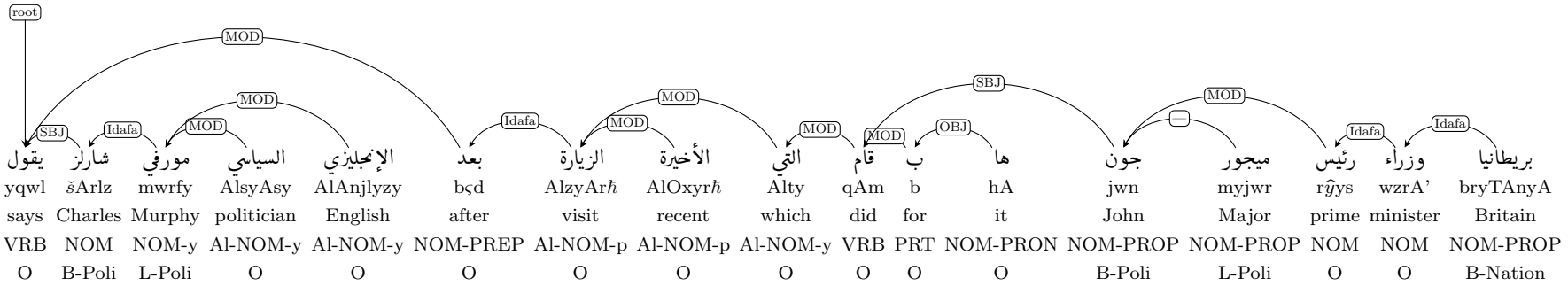

Figure 7.2: The second example of a dependency structure.

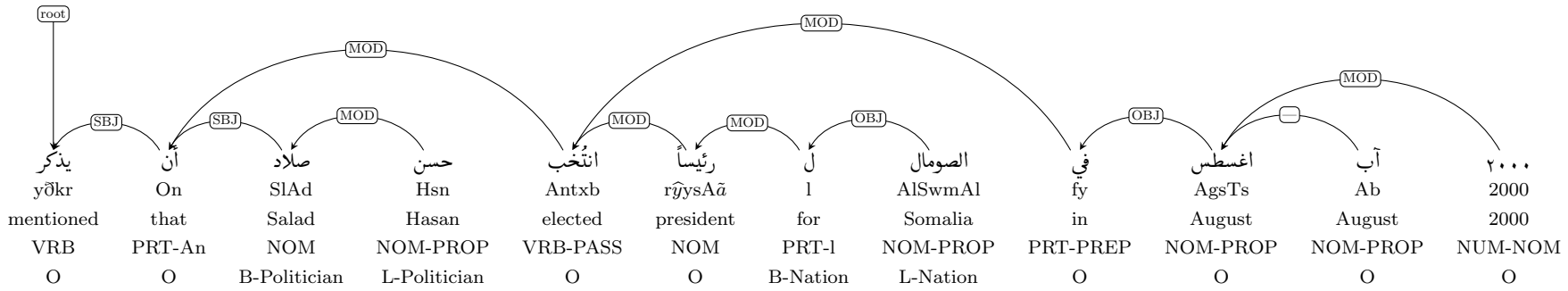

Figure 7.3: The third example of a dependency structure.

7.2.1 Dependency-based Feature set

The representation of the dependency structure presents each token as a node. A particular token (T) should have only one head (H), except for the root, and zero or more dependents (D). A token (T) can have zero or more siblings (S), where they are connected (i.e. are dependent on) to the same head. Therefore, the following set of features as seen in Table 7.1 can be extracted and then be fed into the ML algorithm:

Table 7.1: The dependency-based Feature set. (This example is drawn from the sentence presented in Figure 7.1 and assuming that the current token is (شيخ /šyx/ 'Sheikh'))

Feature	Example
The current token T	شيخ
POS of T	NOM_PROP
The presence of T in the Gazetteer (i.e. WikiFANE$_{Gazet}$)	NA
Syntactical role of T (SUB, OBJ, Idafa, etc.)	MOD
Token of 1st, 2nd and 3rd Head H	{(رئيس /r$\widehat{y}$ys/ 'Head'), (قال /qAl/ 'Said'), root}
Syntactical role of 1st, 2nd and 3rd H	{SUB, NA, NA}
POS of 1st, 2nd and 3rd H	{NOM, VRB, NA}
Token of 1st, 2nd and 3rd Dependent D or 'NA' otherwise	{(شريف /šryf/ 'Sheriff'), NA, NA}
Syntactical role of 1st, 2nd and 3rd D or 'NA' otherwise	{—, NA, NA}
POS of 1st, 2nd and 3rd D or 'NA' otherwise	{NOM_PROP, NA, NA}
Token of 1st, 2nd and 3rd Sibling S or 'NA' otherwise	{(مجلس /mjls/ 'Council'), NA, NA}
Syntactical role of 1st, 2nd and 3rd S or 'NA' otherwise	{Idafa, NA, NA}
POS of 1st, 2nd and 3rd S or 'NA' otherwise	{NOM, NA, NA}

The 1st, 2nd and 3rd 'H' represent the parent, grandparent and great grandparent heads; while the 1st, 2nd and 3rd 'S' represent the first three siblings (if any).

7.2.2 State of the Art Arabic Dependency Parsers

Considering the supervised-based parsers of syntactic representation, i.e. either constituent or dependency structure, there are three main Treebanks for Arabic. A Treebank is a corpus of collection of sentences that has been manually annotated for particular goal, such as syntactic or semantic structure. The Penn Arabic Treebank[2] (PATB) is one of the earliest resources, work commenced on it in the form of a project in 2001. This Treebank is annotated for morphological, phrasal and syntactical information, and provides an English gloss. It has been released in different parts with some variations and it is only available through the LDC. The Prague Arabic Dependency Treebank[3] (PADT) provides richer information than the PATB, including lemma choices. The Stanford parser (Green and Manning, 2010) and the TurboParser (Martins et al., 2010), are examples of parsers producing dependency output for Arabic texts, relying on PATB with some variations. The Columbia Arabic Treebank (CATiB) project, started in 2008, differs from PATB and PADT as it uses a small number of tags to represent dependency (Habash and Roth, 2009). In addition, it clearly annotates two important aspects of Arabic grammatical roles, i.e. Idafa and Tamyiz (TMZ). Tamyiz is a construction that relates two nouns where the first and the second nouns are called specified and specifier. The second noun is always singular in number and accusative in case (Habash, 2010).

CATiB provides a pipeline-based tool using text in a utf-8 format to produce dependency information. The output for each sentence is presented in columns per token as follows: Index \t token \t POS \t head_index \t syntactical_role.

CATiB pipeline produces 44 POS tags (Habash, 2010, p.83). This extended tag set relies on the basic CATiB POS tag set which are:

1. NOM: the non-proper nominal that includes common nouns, adjectives and adverbs.

2. PROP: proper nouns.

[2]The first part of PATB: http://catalog.ldc.upenn.edu/LDC2003T07
[3]PADT: http://catalog.ldc.upenn.edu/LDC2004T23

3. VRB: active-voice verbs.

4. VRB-PASS: passive-voice verbs.

5. PRT: particles including conjunctions and prepositions.

6. PNX: punctuation.

In its syntactical roles, CATiB uses eight relation labels, as follows:

1. SBJ: the subject of verb or topic of simple nominal sentence.

2. OBJ: the object of verb.

3. TPC: the topic in complex nominal sentences containing an explicit pronominal referent.

4. PRD: the predicate marking the complement of the structure of (كان /kAn/ 'was') and (إن /In/ 'that').

5. Idafa: the relationship between the possessor [dependent] and the possessed [head] in the Idafa/possesive nominal construction.

6. TMZ: the relationship of the specifier [dependent] to the specified [head] in the Tamyiz/specification nominal constructions.

7. MOD: the general modifier of verbs or nouns.

8. (−): the flat symbol is a special label given to label constructions such as first-last proper name sequences.

7.2.3 Evaluation

It was decided to use the CATiB pipeline tool[4] (produced by Marton et al. (2013)), to parse all corpora and to produce the set of features presented in the previous section. Since the POS tag set produced using the CATiB pipeline tool is very limited, the dataset has been parsed by CATiB and AMIRA and then the result from both parsers has been merged. The same classifier (CRF) was used, with a similar encoding scheme (i.e. BILOU).

This is shown in Table 7.2 where, in all corpora, the performance of the dependency-based representation alone outperforms that of window-based representation. WikiFANE$_{Auto}$ scores the highest improvement by 4.75% whereas the WikiFANE$_{Gold}$ scores the lowest by 2.34% in this experiment compared with the previous one. Moreover, the recall metrics reveal improvement across corpora, suggesting that the dependency-based representation has the ability to capture an increased number of NEs when compared to the traditional window-based representation.

Table 7.2: The results of the dependency-based features representation. ('+|-' represents the variation compared with the previous experiment)

Corpus	Development			Test			+|-
	P	R	F	P	R	F	
NewsFANE$_{Gold}$	79.84	56.75	66.34	76.14	57.7	65.65	**3.98**
WikiFANE$_{Gold}$	71.17	46.95	56.58	75.18	45.1	56.38	**2.34**
WikiFANE$_{Auto}$	87	73.55	79.71	85.78	69.18	76.59	**4.75**

7.3 Exploiting Hybrid Representation

This experiment evaluates combining both representations, i.e. window- and dependency-based, in one model to obtain the maximum benefit from both approaches. We applied the set of features presented in Section 6.2.2 and 7.2.1 together and used the same classifier and the encoding scheme as shown in previous experiment.

[4]Not yet released to the public. We would like to thank the author for permission for its use.

7.3.1 Evaluation

Thus far, the features involved in the classifier are the one extracted from the dependency structure of the sentence. In the hybrid representation, we investigated merging of window- and dependency-based representations in a single representation. For example, the features extracted from the token '(شيخ /šyx/ 'Shaikh')' in Figure 7.1 is expressed in details in Table 7.3.

The results presented in Table 7.4 demonstrate that the classifier tends to utilise both dependency-based and window-based representations in all corpora efficiently. It is worth noting that NewsFANE$_{Gold}$ and WikiFANE$_{Gold}$, as gold-standard corpora of different genres, reveal notable improvements of 4.03% and 4.63% in the F-measure respectively in the hybrid representation. We notice that both precision and recall have been boosted, which improves the F-measure for all corpora.

7.4 Error Analysis

In this section we analyse the errors in the two experiments presented in this chapter as was done in the previous chapter (see Section 6.6). Since we have two experiments in this chapter, the goal of this evaluation is to show the variation of an individual experiment and to compare it with the one conducted before as follows:

- The experiment presented in Section 7.2.3, which was carried out to evaluate the effect of the dependency-based representation, is compared with the experiment presented in Section 6.5[5] in the previous chapter.

- The experiment presented in Section 7.3.1 is compared with the experiment shown in Section 7.2.3.

The symbol '+|-' is used to represent the difference of the value for each metric when comparing the current experiment with the previous one.

[5]The experiment that used 'BILOU' scheme is selected for this comparison due to it scored the highest.

Table 7.3: The hybrid-based feature set. (This example is drawn from the sentence presented in Figure 7.1 and assuming that the current token is (شيخ /šyx/ 'Sheikh'))

Feature	Example
Dependency-based features	
The current token T	(شيخ /šyx/ 'Shaikh')
POS of T	NOM_PROP
The presence of T in the Gazetteer (i.e. WikiFANE$_{Gazet}$)	NA
Syntactical role of T (SUB, OBJ, Idafa, etc.)	MOD
Token of 1st, 2nd and 3rd Head H	{(رئيس /rʔys/ 'Head'), (قال /qAl/ 'Said'), root}
Syntactical role of 1st, 2nd and 3rd H	{SUB, NA, NA}
POS of 1st, 2nd and 3rd H	{NOM, VRB, NA}
Token of 1st, 2nd and 3rd Dependent D or 'NA' otherwise	{(شريف /šryf/ 'Sheriff'), NA, NA}
Syntactical role of 1st, 2nd and 3rd D or 'NA' otherwise	{—, NA, NA}
POS of 1st, 2nd and 3rd D or 'NA' otherwise	{NOM_PROP, NA, NA}
Token of 1st, 2nd and 3rd Sibling S or 'NA' otherwise	{(مجلس /mjls/ 'Council'), NA, NA}
Syntactical role of 1st, 2nd and 3rd S or 'NA' otherwise	{Idafa, NA, NA}
POS of 1st, 2nd and 3rd S or 'NA' otherwise	{NOM, NA, NA}
Window-based features	
Window of tokens surrounding the current token	{(شيخ /šyx/ 'Shaikh'), (شريف /šryf/ 'Sheriff'), (في /fy/ 'in'), (الصومال /AlSwmAl/ 'Somalia')}
Character-level features (C1)	(ش /š/)
Character-level features (C1 + C2)	(شي /šy/)
Character-level features (C1 + C2 + C3)	(شيخ /šyx/)
Character-level features (Cn-2 + Cn-1 + Cn)	(شيخ /šyx/)
Character-level features (Cn-1 + Cn)	(يخ /yx/)
Character-level features (Cn)	(خ /x/)
Stem of current token	(شيخ /šyx/)
POS of T-1, T-2, T+1, T+2	{NOM_PROP, PRT-PREP, NOM_PROP, NOM_PROP}
BPC of T, T-1, T-2, T+1, T+2	{B-PP, I-PP, I-PP, I-PP}

Table 7.4: The results of the hybrid approach using dependency-based and window-based features representation.

Corpus	Development			Test			+|-
	P	R	F	P	R	F	
NewsFANE$_{Gold}$	82.08	57.77	67.81	80.21	61.58	69.68	**4.03**
WikiFANE$_{Gold}$	89.31	49.11	63.37	83.34	50.48	62.88	**4.63**
WikiFANE$_{Auto}$	87.03	73.29	79.57	87.31	76.17	77.81	**1.22**

7.4.1 Confusion Matrix

The analysis of the confusion matrix for both experiments, i.e. dependency- and hybrid-based, is presented in a single table for each corpus. As evident in Table 7.5, each cell presents two digits separated by '|'. Those digits represent the difference in the values of the confusion matrix between two experiments. The numbers on the left of the '|' represent the difference between the dependency-based experiment (as presented in Section 7.2.3) and the window-based experiment (as presented in Section 6.5), whereas the numbers on the right of the '|' show the difference between the hybrid experiment and the dependency-based one. This method of presentation highlights the variation of the performance of the probabilistic model for each class. In this representation, positive numbers captured in the diagonal cells report positive improvement of the classifier to correctly predict the tags. For other cells, excluding the diagonal ones, negative numbers show that the classifier has fewer struggles among different classes.

For the NewsFANE$_{Gold}$ corpus, for example, the classifier is 41 points more accurate in the assignment of 'PER' NEs in the dependency-based experiment than in the window-based experiment. Moreover, the classifier performance was further improved by having another 18 'PER' NEs correctly tagged in the hybrid-based experiment. This shows a positive improvement when using the dependency-based approach and a further improvement when using the hybrid approach.

As evident in Table 7.5, the dependency-based approach yields improvements across most classes. For example, 'ORG' and 'PRO' NEs have been correctly tagged and thus

the result is improved by 204 and 80 respectively, whereas 'VEH' and 'O' have shown degradation by 8 and 3 entities respectively.

Where the hybrid-based approach is concerned, the classifier still performs positively by tagging the correct classes, except for two cases in 'ORG' and 'PRO' where 4 and 15 NEs respectively were mistakenly tagged. Both experiments show that the classifier has improved its ability to avoid misclassification.

Table 7.5: The variation of the confusion matrix of NewsFANE$_{Gold}$ of the dependency- and hybrid-based experiments. (The '|' separates the difference of the experiments presented in Section 6.5 and Section 7.2.3)

	PER	ORG	LOC	GPE	FAC	VEH	WEA	PRO	O
PER	**41|18**	1|1	0|0	8|-2	0|0	0|0	0|0	1|-1	-51|-16
ORG	-5|2	**204|-4**	0|0	-10|-1	0|0	-1|0	0|0	0|0	-188|3
LOC	0|0	0|0	**2|4**	-1|0	0|0	0|0	0|0	0|0	-1|-4
GPE	5|-3	-1|1	0|0	**13|13**	0|0	0|0	0|0	-1|0	-16|-11
FAC	-1|0	-31|0	0|0	-3|-1	**52|0**	0|0	0|0	0|0	-17|1
VEH	0|1	4|-2	0|0	0|0	0|0	**-8|3**	0|0	0|0	4|-2
WEA	0|0	0|0	0|0	0|0	0|0	0|0	**0|0**	0|0	0|0
PRO	-2|0	-2|0	0|0	-2|-1	-1|0	0|0	0|0	**80|-15**	-73|16
O	16|-12	-14|-7	0|0	-4|0	2|-2	-1|0	0|0	4|-3	**-3|24**

Table 7.6 shows the variation of the confusion matrix of WikiFANE$_{Gold}$ of the dependency- and hybrid-based experiments. In the dependency-based experiment, we notice degradation in 'PER', 'GPE', 'FAC' and 'VEH' by 271, 184, 9 and 4 respectively. On the other hand, 'ORG', 'LOC', 'PRO' and 'O' improved by 49, 4, 18 and 300 respectively.

For the hybrid-based experiment, the 'PER' class has overcome the degradation of the previous experiment and risen by 328 while 'GPE's negative assignment has increased by 26. Other classes show improvement. For example, 'ORG' has increased by correctly tagging 236 entities.

Table 7.7 shows the variation of the confusion matrix of WikiFANE$_{Auto}$ of the dependency- and hybrid-based experiments. For both experiments, the classifier performed positively in assigning the correct tags. For example, the classifier scores improved by 444 and 101 in tagging 'GPE' entities in dependency- and hybrid-based experiments respectively. There

Table 7.6: The variation of the confusion matrix of WikiFANE$_{Gold}$ between window-, dependency-, and hybrid-based experiments

	PER	ORG	LOC	GPE	FAC	VEH	WEA	PRO	O
PER	**-271|328**	-3|-3	-4|0	-18|6	-1|-1	0|0	0|0	0|-2	297|-328
ORG	8|-6	**49|236**	-2|0	5|-4	-14|6	1|-2	0|0	-1|-2	-46|-228
LOC	-3|1	0|3	**4|41**	-20|-5	-1|-3	0|0	0|0	0|0	20|-37
GPE	-4|-9	-1|-1	-40|39	**-184|-26**	-2|0	0|0	0|0	-2|0	233|-3
FAC	-2|0	0|0	4|-7	1|1	**-9|76**	-2|2	0|0	0|0	8|-72
VEH	0|4	0|0	0|0	0|2	0|0	**-4|28**	0|0	0|0	4|-34
WEA	0|0	0|0	0|0	0|0	0|0	0|0	**0|0**	0|0	0|0
PRO	2|-3	0|0	0|0	-1|0	-2|0	-2|0	0|0	**18|53**	-15|-50
O	-119|-33	-39|-7	-29|-6	-74|-19	-17|-4	-9|0	0|0	-13|-3	**300|72**

were some cases where the classifier mistakenly tagged non-NEs into one of the classes. For example, the classifier mistakenly assigned the tag 'PER' for 101 tokens where they should have been tagged as 'O'.

Table 7.7: The variation of the confusion matrix of WikiFANE$_{Auto}$ between window-, dependency-, and hybrid-based experiments

	PER	ORG	LOC	GPE	FAC	VEH	WEA	PRO	O
PER	**81|30**	1|0	-1|2	8|-12	4|-3	0|0	0|0	0|0	-93|-17
ORG	-2|0	**19|26**	0|0	-3|0	0|0	0|0	0|0	0|0	-14|-26
LOC	-4|0	0|0	**3|46**	-2|-1	0|0	0|0	0|0	0|0	3|-45
GPE	11|-11	0|0	-2|0	**444|101**	0|0	0|0	0|0	0|0	-453|-90
FAC	3|3	0|0	-1|0	0|4	**36|-40**	0|0	0|0	0|0	-38|33
VEH	0|0	0|0	0|0	0|0	0|0	**6|-2**	0|0	0|0	-6|2
WEA	0|0	0|0	0|0	0|0	0|0	0|0	**0|0**	0|0	0|0
PRO	0|0	-1|0	0|0	-1|0	0|0	0|0	0|0	**50|5**	-48|-5
O	101|5	14|-5	7|-2	163|-49	1|-6	0|0	0|0	4|-2	**-290|59**

7.4.2 NEs Phrase Length

Table 7.8 summarises the error analysis of the NEs phrase length across all corpora for both experiments. We use '+|-' to present the variation of the result of one experiment when compared with the previous one.

Although the performance of the classifier over NewsFANE$_{Gold}$ degraded by 10% in terms of detecting single-word NEs in the dependency-based experiment when compared

to the window-based experiment presented in Section 6.5, the same classifier showed an improvement of 28% in multi-word NEs. This indicates that the dependency-based approach captures useful benefits in the NewsFANE$_{Gold}$. This is not the case with WikiFANE$_{Gold}$ and WikiFANE$_{Auto}$ where the improvements were only 1% and 5% respectively. In the same experiment, we noticed that WikiFANE$_{Auto}$ improved by 12% at the single-word level.

In the hybrid-based experiment, NewsFANE$_{Gold}$ and WikiFANE$_{Auto}$ showed little variation when compared with the previous experiment. On the other hand, WikiFANE$_{Gold}$ score improved by 23% in the multi-word level.

This analysis shows that the two approaches have different strengths particularly in terms of the length of the NE.

Table 7.8: Error analysis in terms of length of NE phrases for dependency- and hybrid-based experiments across all corpora. '+|-' represents the difference of the current experiment when compared with the previous one.

Error on phrase length	%	+|-	%	+|-
	NewsFANE$_{Gold}$			
	Dependency-based		Hybrid-based	
Single-word	49%	10%	45%	-4%
Multi-word	25%	-28%	23%	-2%
	WikiFANE$_{Gold}$			
	Dependency-based		Hybrid-based	
Single-word	62%	16%	70%	8%
Multi-word	44%	-1%	21%	-23%
	WikiFANE$_{Auto}$			
	Dependency-based		Hybrid-based	
Single-word	31%	-12%	31%	0.31
Multi-word	27%	-5%	24%	0.24

7.4.3 Fine-grained Classes of the Same Parent

The evaluation of the classifier's ability to distinguish between fine-grained classes that share the same parent is presented in Table 7.9. We summarise the results for all corpora across both experiments. In this table, we present the percentage of the error that shows

how many times the classifier failed to distinguish between fine-grained classes of the same parent. The variation when compared with the previous experiment is also presented using the '+|-' symbols. A negative variation result means less error when compared with the previous experiment.

For the dependency-based experiment, as evident in Table 7.9, the ability for the classifier to correctly classify fine-grained classes was improved across all corpora. NewsFANE$_{Gold}$ showed most improvement, having reduced the error by 7.96%. WikiFANE$_{Gold}$ and WikiFANE$_{Auto}$ also showed a slight improvement of 1.21% and 0.05% respectively. The reduction of errors shows that the dependency-based approach is able to detect useful clues from the dependency structure which allows for correct classification of fine-grained classes of the same parent.

In the hybrid-based experiment the classifier continued to show improvement over two corpora. WikiFANE$_{Gold}$ and NewsFANE$_{Gold}$ improved by 3.5% and 0.69% while WikiFANE$_{Auto}$ score decreased by 0.32%.

Table 7.9: Error analysis of tagging fine-grained NEs that share same parent for dependency- and hybrid-based experiments across all corpora. '+|-' represents the difference of the current experiment when compared with the previous one.

	NewsFANE$_{Gold}$	
	Dependency-based	Hybrid-based
%	6.13%	5.44%
+|-	-7.96%	-0.69%
	WikiFANE$_{Gold}$	
	Dependency-based	Hybrid-based
%	6.86%	3.36%
+|-	-1.21%	-3.50%
	WikiFANE$_{Auto}$	
	Dependency-based	Hybrid-based
%	0.94%	1.26%
+|-	-0.05%	0.32%

7.5 Chapter Summary

In this chapter, we presented a new methodology to represent the features by relying on dependency structure. Section 7.1 started off by discussing the limitations of the traditional window-based features representation. In Section 7.2 we presented the usefulness of relying on the dependency structure where NER is concerned. In the following section, we combine both representations, i.e. the window- and the dependency based representation, into one model. The chapter ends by analysing the errors made in both experiments.

So far, all methods and experiments presented in Chapter 6 and 7 focus on capturing evidences (i.e. clues) in context and sentence level. In the following chapter, we will present our method of capturing global evidence that go beyond the sentence boundary in order to reduce the data sparsity. In this approach we relied on clustering technique of large unannotated textual data to achieve this goal.

Chapter 8

Exploiting Global Evidence

Chapter Synopsis

The previous chapter presented a new approach that relies on the dependency structure in order to overcome the limitations of window-based representation. However, this chapter will discuss the supplementary methodology to exploit the unannotated textual data in order to capture global evidence to increase the performance of the fine-grained NER (Section 8.1), followed by examination of an experiment conducted across different corpora to evaluate the proposed methodology by relying on the Brown clustering algorithm (Section 8.2). Finally, Section 8.3 analyses the error according to different metrics.

8.1 Capturing Global Evidence

8.1.1 The Intuition

Thus far, this thesis has examined the window-based and dependency-based representations of evidence (as presented in Chapters 6 and 7 respectively), in order to increase the performance of the classification process. However, there is still room for improvement, since both of these approaches focus only at the sentence level. This chapter will investigate the approach to capturing global evidence beyond the sentence level.

Currently, virtually all published studies, founded on the author's best knowledge, on the subject of the Arabic NER apply the predefined window-based representation as

examples when using this approach (Shaalan and Oudah, 2014; Benajiba et al., 2009b). In relation to English, Ratinov and Roth (2009) implemented two ways of capturing global evidence. The first approach was 'context aggregation', which works by searching the entire document for a given token and returning the context of size two around each matched token. For the purposes of the study, Ratinov and Roth (2009) limited the search to within 200 tokens. The second approach was 'extended prediction history', which captures the 1000 previous tokens and counts the frequency of the label of the target class.

Another means of achieving this, which has not yet been investigated for Arabic NER, is by utilising unannotated textual data, by clustering tokens into semantic groups based on context similarity. The reasoning behind this approach is that an NE token such as (الطائف /AlTAf/ 'Taif')[1], which is not seen in the training process, cannot be correctly classified, since it contains neither window-based nor dependency-based evidence in the training phase. Performing clustering over unannotated textual dataset results in putting (الطائف /AlTAf/ 'Taif') and (لندن /lndn/ 'London') in the same cluster because they appear in similar context in unannotated text several times. Thus, this sort of knowledge increases the capacity of the classifier to a global level that go beyond the sentence boundary.

With regard to NER, data sparseness is a common problem in supervised machine learning approaches (Allison et al., 2006; Lafferty et al., 2001). The problem occurs when a probabilistic model is expected to tag a certain input where this has not been seen in the training phase. The sparsity of data becomes salient for Arabic due to the complex morphological structure of the word formation (Goweder and De Roeck, 2001; Benajiba et al., 2007; Meftouh et al., 2008). However, one way to reduce the sparsity of data

[1]Taif is a city in Saudi Arabia

is to apply a word clustering technique on extensive unannotated textual data, thereby reducing the dimensions of the features by mapping back the clustering output as features in the probabilistic model (Liang, 2005).

8.1.2 An Overview of Brown Clustering

Among various word clustering techniques such as distributed word embeddings (Bengio et al., 2006; Collobert and Weston, 2008; Mnih and Hinton, 2009) and dimensionality reduction (Lamar et al., 2010; Deerwester et al., 1990), the Brown clustering algorithm favours alternative approaches due to its simplicity, the hierarchical nature of the output and the implementation availability (Liang, 2005; Stolcke et al., 2002), which suit the goal of this study.

The Brown clustering algorithm is a class-based bigram language model that works by maximising the mutual information of adjacent clusters (Brown et al., 1992; Liang, 2005; Šuster and Van Noord, 2014). It uses a hierarchical representation for the clusters, meaning that the word class can be chosen at different levels in the hierarchy. The key idea underpinning the Brown clustering algorithm is that words that share similar clusters have a similar distribution to neighbouring words. For example, the words (السبت /Alsbt/ 'Saturday') and (الأحد /AlOHd/ 'Sunday') are expected to belong to the same cluster, because they share a similar distribution of the words that come immediately before and after them within several contexts.

Technically, the Brown clustering algorithm takes a sequence of words $(w_1, ..., w_n)$ as an input and produces a binary tree, i.e. clusters, as an output where the leaves of the tree are the words. An example of the output of the Brown algorithm is illustrated in Figure 8.1.

The bit string id starting from the root down to the leaf forms the clustering identification. For example, the words 'Apple and IBM' reside in the cluster '010', whereas 'Toyota, Ford and Volvo' belong to the cluster '011'. Moreover, capturing the upper level, i.e. internal nodes, of the bit string in the tree such as cluster '01', results in the inclusion

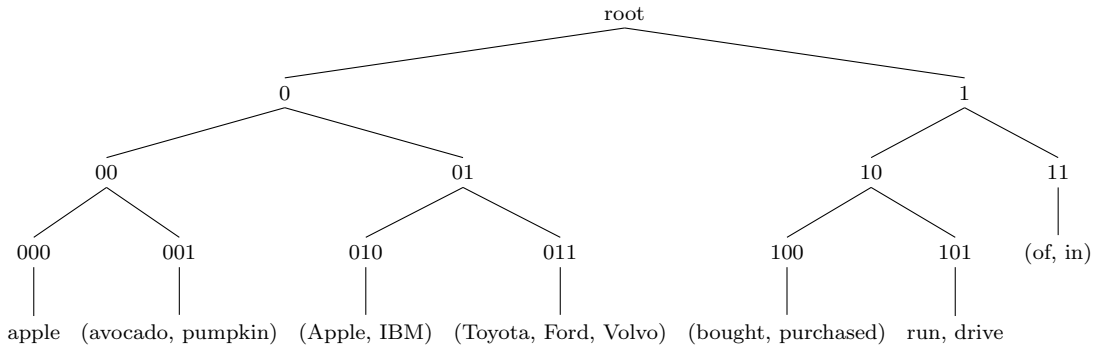

Figure 8.1: An illustrated example of the output of the Brown clustering algorithm (Šuster and Van Noord, 2014)

of all the words mentioned above as being in clusters '010' and '011' within one cluster.

Herein, we present a short contextual background to the Brown algorithm and more mathematical details as presented in (Brown et al., 1992; Liang, 2005).

Assuming that:

- V is the set of the words of the corpus
- $w : w_1, w_2, ..., w_m$ is the word sequence with $w \in V$

If, $C : V \rightarrow 1, 2, ..., k$ is a partition function of the vocabulary into k classes, the likelihood model is defined as:

$$P(w; C) = \prod_{i=1}^{m} p\,(\,w_i \,|\, C(\,w_i\,)\,)\;.\;p\,(\,C(\,w_i\,)\,|\,C(\,w_i - 1\,)\,) \tag{8.1.1}$$

As derived by Brown et al. (1992), Equation 8.1.1 can be written down in a more convenient way as:

$$logP(w; C) = \sum_{i=1}^{m} log[\,p\,(\,w_i \,|\, C(\,w_i\,)\,)\;.\;p\,(\,C(\,w_i\,)\,|\,C(\,w_i - 1\,)\,)] \tag{8.1.2}$$

Defining the quality of the clustering, Liang (2005) viewed the clustering process in the context of a class-based bigram language model, as shown in Figure 8.2.

Therefore, the quality of clustering C that maps each word to a cluster is defined as the logarithm of this probability (see Equation 8.1.3) normalised by the length of the text:

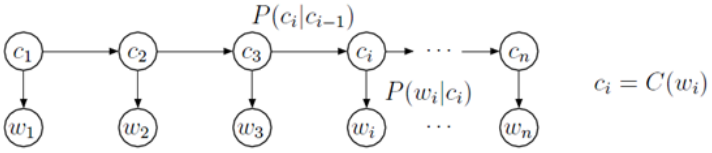

Figure 8.2: An illustration of the class-based bigram language model, which defines the quality of a clustering, represented as a Bayesian network (Liang, 2005).

$$Quality(C) = \frac{1}{m}\sum_{i=1}^{m} log[p(w_i|C(w_i)) \cdot p(C(w_i)|C(w_i-1))] \tag{8.1.3}$$

8.1.3 Inducing NER by Clustering Knowledge

The hierarchal representation of the Brown clustering algorithm facilitates the inclusion of different semantic levels of granularity. The output from the clustering delivers valuable information, which can be utilised by the NER system. This information can be divided into three categories:

1. The cluster of tokens belongs to the NE category. For example, and illustrated in Figure 8.3, (شيكاغو /šykAγw/ 'Chicago') and (طوكيو /Twkyw/ 'Tokyo') belong to the same cluster, where both are NE type 'LOC'. In addition, (هديل /hdyl/ 'Hadeel') and (ممدوح /mmdwH/ 'Mamdooh') fall into similar clusters, and are both 'PER' NE.

2. The cluster of keyword tokens provides an informal insight into the target NE classes. For example, (كتائب /ktAǔb/ 'Brigades') and (منظمة /mnĎmħ / 'Organisation') are keywords which infer the context of 'ORG' NE. The context is expressed, for instance, as (كتائب شهداء الأقصى /ktAǔb šhdA' AlOqSý/ 'Al Aqsa Martyrs

Brigades') or (منظمة العفو الدولية /mnǍmħ Alʕfw Aldwlyħ / 'Amnesty International'). Both head tokens in the NEs refer to the same cluster, which indicates 'ORG' (see Figure 8.3).

3. The cluster of the head and dependent tokens that the current token is pointing to. For example, the token (شيخ /šyx/ 'Shaikh'), as shown in Figure 8.4, points to the head token (رئيس /rў̆ys/ 'President') where the 'رئيس' belongs to the cluster '1110000111'. This clustering knowledge enables the building of an abstract semantic representation for tokens. This implies that the token 'رئيس' can be replaced as (مدير /mdyr/ 'Manager') in other sentences where both tokens belong to the same cluster.

Further examples are presented in Figure 8.3, where the group and the cluster id headings refer to name and cluster respectively.

Figure 8.3: Examples of the output of the Brown algorithm when applied to Arabic textual data. The group column represent the following: (A): First names, (B): Last names, (C): Locations, (D): Organisational keywords, (E): Locational keywords, and (F): Facility-related keywords.

Group	Cluster id	Examples
A	000011111111101	(هديل /hdyl/ Hadeel) (حميدان /HmydAn/ Homaidan) (ممدوح /mmdwH/ Mumdooh)
B	000011000101 000011000110	(الساهر /AlsAhr/ Alsaher) (البخاري /AlbxAry/ Albokhari) (الحازمي /AlHAzmy/ Alhazmi)
C	0101101100	(بكين /bkyn/ Beijing) (تكساس /tksAs/ Texas) (طوكيو /Twkyw/ Tokyo)
D	0111111111111011000	(كتائب /ktAў̆b/ battalions) (جبهة /jbhħ/ front) (منظمة /mnD̆mħ/ organization)
E	011110110000	(مستوطنة /ktAў̆b/ settlement) (ضاحية /DAHyħ/ suburb) (محمية /mHmyħ/ protectress)
F	101101100111011	(استاد /AstAd/ stadium) (جسر /jsr/ bridge) (مطار /mTAr/ airport)

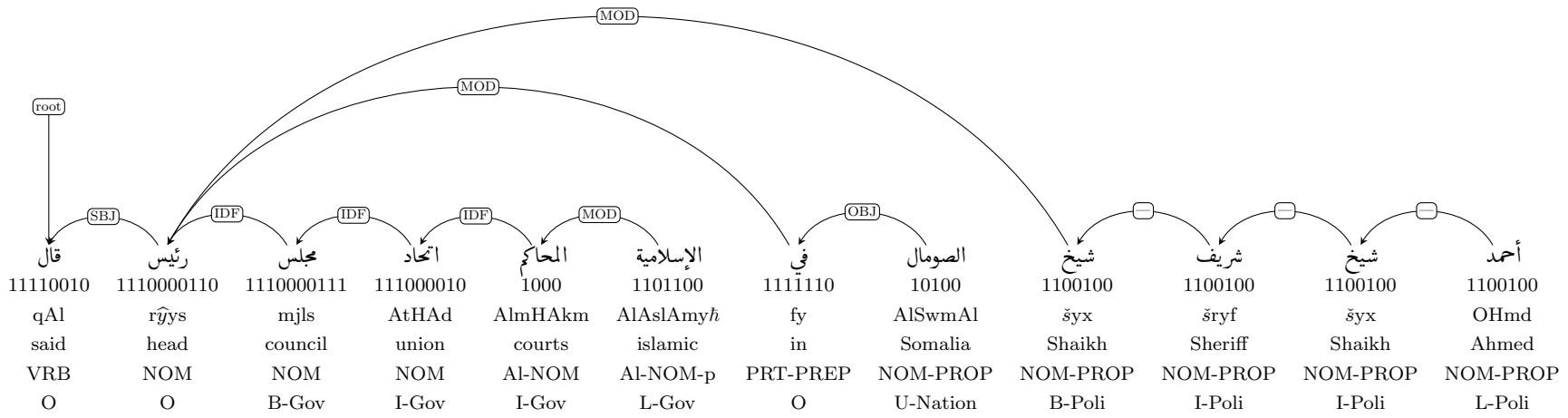

Figure 8.4: An example of the dependency structure of Arabic sentences. The second row represents the clusters according to the Brown algorithm (the sentence is displayed left to right).

8.2 Evaluation

In this section, we conducted an experiment over three corpora, namely NewsFANE$_{Gold}$, WikiFANE$_{Gold}$ and WikiFANE$_{Auto}$, in order to evaluate the proposed approach.

8.2.1 Source of the Data

The aim of this experiment was to evaluate the usefulness of injecting the clustering information from the Brown algorithm into the supervised model. However, the actual size of the corpora as cited in Chapter 5 was too small to apply the Brown algorithm. Therefore, an alternative set of different unannotated corpora, of a reasonably large size from different sources, was prepared for use in this experiment, as demonstrated in Table 8.1.

Table 8.1: Different textual data used in the Brown algorithms

Source of unannotated dataset	Size	Used for
NewsFANE$_{Gold}$ + Gigaword	1.17M	NewsFANE$_{Gold}$
WikiFANE$_{Gold}$ + WikiFANE$_{Auto}$	2.5M	WikiFANE$_{Gold}$
WikiFANE$_{Gold}$ + WikiFANE$_{Auto}$	2.5M	WikiFANE$_{Auto}$

The first and second columns in Table 8.1 show the source of the unlabelled textual data used in the Brown algorithm and the respective size of the data. The final column shows the target corpus using the knowledge in the supervised model.

Random articles were selected from Arabic Gigaword[2] (Parker et al., 2011), as well as textual data from NewsFANE$_{Gold}$, to create unannotated data of 1.17M tokens. The Gigaword subset was selected due to the similarity of its genre to NewsFANE$_{Gold}$ as they are newswire-based corpora. The textual data for both WikiFANE$_{Gold}$ and WikiFANE$_{Auto}$[3] were collated into one data set in order to generate clustering knowledge for both WikiFANE$_{Gold}$ and WikiFANE$_{Auto}$.

A tokenisation and preprocessing step was conducted on different textual dataset in order to achieve tokenisation scheme compatibility between those datasets and the one

[2]I'd like to thank Professor Martin Russell (School of Electronic, Electrical and Computer Engineering) for requesting a copy of the Arabic Gigaword Fourth Edition corpus from LDC

[3]In this experiment the whole textual data of this corpus is used as presented in Section 5.1.6

used in the supervised model. The AMIRA toolkit (Diab, 2009) was used with the tokenisation scheme of (conj+prep+suff)[4].

8.2.2 Extracting Clustering Features

The Brown algorithm[5] was run in order to cluster the tokens into 1000 clusters, as suggested by Miller et al. (2004); Liang (2005); Ratinov and Roth (2009); Tkachenko et al. (2012). The output of the Brown algorithm (which involves 1000 clusters) was injected as a set of features by extracting the clustering bit string of (4, 6, 8, 10, 12) in a similar way to the research presented by Turian et al. (2010); Tkachenko et al. (2012). The reasoning behind this representation of the output was to facilitate a flexible level of grouping tokens into semantic clusters. For example, the tokens (البخاري /AlbxAry/ 'Albokhari') and (الحازمي /AlHAzmy/ 'Alhazmi') are clustered into '000011000101' and '000011000110' respectively, where both are 'PER' NE. They share the first 10 bits of the cluster, and this level of granularity allows for the extraction of useful knowledge to classify both tokens into the same class.

8.2.3 The Result

In this experiment, we used the same supervised ML classifier, i.e. CRF, as in the previous experiment. The set of the features used are those presented in Section 7.2.1 and the clustering features presented in Section 8.2.2.

The result of the experiment is shown in Table 8.2, which demonstrates that notable improvement was achieved across all corpora. WikiFANE$_{Auto}$ scores the highest F-measure, although all other corpora gained improvements. The results reveal that the recall sharply improved by 7 and 10 points for NewsFANE$_{Gold}$ and WikiFANE$_{Gold}$ respec-

[4]In this scheme the conjunctions, prepositions and suffixes are separated by white space.
[5]We relied on the implementation of Liang (2005) for the Brown clustering algorithm.

tively. This implies that injecting the Brown clusters improved the recall metric as a means of having the ability of delimiting an increased number of NEs.

Table 8.2: The results of injecting the output of Brown clustering into the CRF model

Corpus	Development			Test			+|-
	P	R	F	P	R	F	
NewsFANE$_{Gold}$	86.13	70.38	77.46	81.66	68.36	74.42	4.74
WikiFANE$_{Gold}$	77.8	62.36	69.23	79.87	60.19	68.64	5.76
WikiFANE$_{Auto}$	89.17	74.04	80.9	88.64	73.18	80.17	2.36

8.3 Error Analysis

The following error analysis was conducted in order to evaluate the differences between the performances of the approach presented in this chapter, which utilised the Brown clustering algorithm, compared with the approach presented in the previous chapter in Section 7.3, which utilised the dependency- and window-based features. In this way, we will be able to capture the variation in both approaches.

8.3.1 Confusion Matrix

In the case of NewsFANE$_{Gold}$, as illustrated in Table 8.3, the classifier gained improvement across almost all classes. For example, there are eighty-one increases in correctly assigning the class 'PER'. The classes 'LOC' and 'WEA' reveal no changes compared with the previous experiment.

The classification process across the WikiFANE$_{Gold}$ data, as seen in Table 8.4, shows misclassification in classes 'ORG', 'FAC', 'WEA', 'PRO' and 'O'. Conversely, the classifier shows improvement over 'PER', 'LOC' and 'GPE' by 179, 9 and 459 times respectively.

The performance over WikiFANE$_{Auto}$, on the one hand, shows improvement in some classes, including 'LOC' by 28 times and 'GPE' by 116 times. On the other hand, it reveals misclassifications on classes such as 'PER' (38 times) and 'PRO' (18 times), as illustrated in Table 8.5.

Table 8.3: The variation of the confusion matrix of NewsFANE$_{Gold}$ between the experiment conducted in this chapter and the previous one.

	PER	ORG	LOC	GPE	FAC	VEH	WEA	PRO	O
PER	**81**	-4	0	-5	-1	0	0	0	-71
ORG	-2	**73**	0	-9	0	0	0	0	-62
LOC	0	0	**0**	0	0	0	0	0	0
GPE	-3	-1	0	**50**	0	0	0	0	-46
FAC	-3	-2	0	2	**9**	0	0	0	-6
VEH	-1	1	0	0	0	**14**	0	0	-14
WEA	0	0	0	0	0	0	**0**	0	0
PRO	0	1	0	1	0	0	0	**16**	-18
O	-11	-4	0	-2	-1	0	0	-2	**20**

Table 8.4: The variation of the confusion matrix of WikiFANE$_{Gold}$ between the experiment conducted in this chapter and the previous one.

	PER	ORG	LOC	GPE	FAC	VEH	WEA	PRO	O
PER	**179**	4	-1	8	2	0	0	7	-199
ORG	3	**-11**	0	2	6	4	0	3	-7
LOC	-3	-2	**9**	15	0	0	0	0	-19
GPE	6	2	10	**459**	1	0	0	0	-478
FAC	1	0	1	5	**-19**	0	0	0	12
VEH	1	0	0	-1	0	**-1**	0	0	1
WEA	0	0	0	0	0	0	**0**	0	0
PRO	1	0	1	1	3	2	0	**-3**	-5
O	25	1	4	29	1	1	0	-1	**-60**

Table 8.5: The variation of the confusion matrix of WikiFANE$_{Auto}$ between the experiment conducted in this chapter and the previous one.

	PER	ORG	LOC	GPE	FAC	VEH	WEA	PRO	O
PER	**-38**	-1	0	7	1	0	0	0	31
ORG	0	**-7**	0	-1	0	0	0	0	8
LOC	0	0	**28**	0	0	0	0	0	-28
GPE	2	2	2	**116**	0	0	0	0	-122
FAC	4	0	0	-3	**0**	0	0	0	-1
VEH	1	0	0	0	0	**0**	0	0	-1
WEA	0	0	0	0	0	0	**2**	0	-2
PRO	0	1	0	0	0	0	0	**-18**	17
O	-15	5	13	-68	5	0	0	0	**60**

8.3.2 Phrase Length of NEs

Injecting the output of the Brown clustering algorithm into the supervised model demonstrates the usefulness of detecting single-word NEs. The errors have been reduced by 3, 19 and 6 for NewsFANE$_{Gold}$, WikiFANE$_{Gold}$ and WikiFANE$_{Auto}$ respectively. Moreover, the performance also carried over onto multi-word NEs in the case of NewsFANE$_{Gold}$. However, the experiments with the WikiFANE$_{Gold}$ and WikiFANE$_{Auto}$ revealed that on the level of multi-word NEs, the performance slightly degraded, as illustrated in Table 8.6.

Table 8.6: Error analysis in terms of length of NEs across all corpora. '+|-' represents the difference between the current experiment compared with the previous one. (Negative values of +|- indicate a reduction in the error level)

Metric	NewsFANE$_{Gold}$		WikiFANE$_{Gold}$		WikiFANE$_{Auto}$	
	%	+|-	%	+|-	%	+|-
Single-word:	42	-3	51	-19	25	-6
Multi-word:	16	-7	24	3	26	2

8.3.3 Fine-grained Classes of the Same Parent

Nevertheless, the importance of this metric is to evaluate how well the classifier performs in classifying fine-grained classes that share the same parent. Table 8.7 summarises the difference between the experiment presented in this chapter and the previous experiment presented in Section 7.3, in relation to all corpora. The symbol '%' is used to illustrate the percentage of the errors, demonstrating how frequently the classifier failed to distinguish between fine-grained classes of the same parent. The variation when compared with the previous experiment is also represented, using the '+|-' symbols. Negative (-) variation indicates that there was a lower error rate than in the previous experiment.

Classification of fine-grained classes of the same parent over NewsFANE$_{Gold}$ and WikiFANE$_{Auto}$ demonstrated improvements, since the error reduced by 0.66% and 0.29% respectively. Whereas, the experiment over WikiFANE$_{Gold}$ shows an increase in the error rate of 1.92%. Table 8.7 illustrates these results.

Table 8.7: Error analysis of tagging fine-grained NEs that share the same parent. '+|-' represents the difference in variation of the current experiment when compared with the previous one.

	NewsFANE$_{Gold}$	WikiFANE$_{Gold}$	WikiFANE$_{Auto}$
%	4.78%	5.28%	0.97%
+|-	-0.66%	1.92%	-0.29%

8.4 Chapter Summary

In this chapter, the methodology of exploiting the global evidence from unannotated textual data has been presented. The Brown clustering algorithm, and how it can be exploited in the NER task, was presented in Section 8.1. This was followed by the evaluation in Section 8.2, where specification of the source of data and clustering utilisation was discussed. In Section 8.3, similar error analysis methodology was conducted in order to closely evaluate the variation of the overall performance of the proposed methodology. Subsequently, the following chapter will present the conclusion of this thesis and recommendations and guidelines for work that could potentially be carried out in future to advance research in the area of Arabic NER.

Part V

CONCLUSION

Chapter 9

Conclusion and Future Work

This thesis addresses the task of fine-grained NER in Arabic, which is very important to other NLP tasks, such as Question Answering (Fleischman and Hovy, 2002; Mollá et al., 2006), Ontology Population (Lee et al., 2006) and Topic Detection (Ng et al., 2007) among others. Fine-grained NER is more challenging than traditional (coarse-grained) NER, in which a large number of semantic classes is involved. Consequently, new challenges arise and new ways to overcome those challenges are necessary. Examples of some of those challenges include the creation of fine-grained resources, investigating appropriate ML techniques, and representing and extracting new features in a suitable manner that transcends traditional approaches.

In this final chapter, the four research questions addressed within this thesis will be revisited:

1. How can annotated fine-grained NE resources, such as corpora and gazetteer be created, to enable supervised fine-grained NER?

2. Which machine learning method is the most efficient in implementing fine-grained NER system?

3. How can informative features that go beyond the local context be defined and extracted, whilst also capturing the semantic differences between fine-grained classes?

4. How can global evidences that goes beyond the sentence boundary be captured in order to enhance the performance of fine-grained NER?

The current thesis has extended the state-of-the-art methodologies to develop NER for Arabic in three angles as follows:

1. Extracted Semantic Knowledge: The current efforts of Arabic NER have only focus on very limited number of coarse-grained classes and been restricted to newswire domain, such as Benajiba et al. (2009a); Darwish (2013); Morsi and Rafea (2013). This thesis considers a deeper semantic tagset by relying on hierarchy-based taxonomy with the coverage of 50 fine-grained classes.

2. Resource Development: Developing the required resources such as corpora and gazetteer has been accomplished manually by recruiting human annotators or by relying on crowdsourcing. This is costly and time-consuming task. Instead, we push the research by presenting a methodology to create annotated corpora and gazetteer automatically with very little human intervention. This has been accomplished by exploiting the richness and the accessibility of the Arabic Wikipedia.

3. Features Representation: The traditional methodology to represent the features is by merely relying on the window-based representation, i.e. words n-gram representation, where the decision made for a token at position (i) is affected by the adjacent two tokens. This methodology has a limitation in which the informative features are only restricted to the size of the window. In this thesis, instead, we advance the research to consider further features that go beyond the size of the window. Therefore, the dependency structure of the sentence has been utilised in order to implement the dependency-based representation.

The following sections will present the main thesis results, contributions and the recommended future work.

9.1 Main Thesis Results

This thesis addresses the task of examining fine-grained NER for Arabic from different angles, with the aim of contributing to various areas.

Relying on ML technology to develop fine-grained NER for Arabic requires the creation of suitable resources, such as annotated corpora and lexical resources. As is shown in Chapter 4, a methodology is devised to develop a scalable gazetteer by exploiting the richness of Arabic Wikipedia.

In order to achieve this, the task was formulated as a document classification problem, requiring the assignment of each Wikipedia article to one of a predefined set of fine-grained classes. A prerequisite of this project is the creation of a taxonomy of 50 fine-grained classes, which this study defines. The taxonomy consists of two of levels, coarse- and fine-grained. The coarse-grained level consists of eight semantic classes: Person, Organisation, Location, Geopolitical Location, Facility, Weapon, Vehicles and Products. The fine-grained level consists of 50 classes (see Table 4.1).

To classify the Arabic Wikipedia articles into fine-grained classes, an investigation aimed to answer the following questions were undertaken:

1. What probabilistic model (i.e. classifier) is suitable for this task (i.e. document classification task)?

2. How should the features be represented, in order that they can be used by the classifier?

3. What is the informative set of features that can be extracted from the Wikipedia article, to be used by the classifier?

In order to answer the first question in systematic way, four probabilistic models, in other words classifiers, are used to evaluate the performance of each one, with different features representation and sets. The classifiers used are: Naïve Bayes (NB), Multinomial Naïve Bayes (MNB), a linear Support Vector Machine (SVM) and Logistic Regression (LR).

Moreover, three possible feature representations are investigated: Term Presence (TP) simply counts the presence of the tokens in the document; Term Frequency (TF) represents how many times the tokens are found in a corpus; and, Term Frequency-Inverse Document Frequency (TF-IDF) reveals how important a given token is to a document within the corpus.

In addition, four sets of features are engineered and extracted: Simple Features (SF) represent the raw dataset, as a simple bag of words without further processing; Filtered Features (FF) represent the dataset after several filtering steps have been taken (including the removal of punctuation, stop words and normalising digits), Language-dependent Features (LF) report the usefulness of the stem representation of the token; and, Enhanced Language-dependent Features (ELF) represent linguistic features, including tokenisation, assigning the POS to each token and distinguishing the tokens based on their location on the Wikipedia page.

In order to identify the different classifiers for this task, a set of 4000 Arabic Wikipedia articles were collated and manually annotated according to two levels of granularities, coarse- and fine-grained. The experiment was first performed at the coarse-grained level, as a pilot experiment. The second experiment was conducted to examine the fine grained classes. Finally, whole Wikipedia articles were fed into the classifier. As a result, a fine-grained gazetteer was developed, with 68355 entities, requiring very little human intervention. It was decided that this resource will be available to the public.

The main findings of Chapter 4 are summarised as follows:

1. Wikipedia, as a public-access resource, can be exploited to develop a scalable fine-grained gazetteer, namely WikiFANE$_{Gazet}$.

2. From a technical perspective, the careful selection of the 3-tuple (Feature Representation, Features Set and Statistical Model) yields significant benefits in the sense of overall classification.

The ability to classify Arabic Wikipedia articles into fine-grained classes facilitates the

automatic projection of the classification results onto the textual data of the Wikipedia articles, in order to develop a scalable annotated corpus, called WikiFANE$_{Auto}$. Wikipedia articles use URL links to link articles together; the visible text of those links is the actual title of the Wikipedia article they link to. Therefore, where one is able to determine what the fine-grained NE type is for each article, a 2-tuple list can be compiled: <the article's title, fine-grained NE type>. Therefore, chapter 5 is dedicated to explaining this idea in detail, towards developing an automated scalable fine-grained Arabic NER corpus. Close examination reveals an important issue; when users write a Wikipedia article they tend to add URL links to phrases that map to another article, at their first mention; however, successive mentions have no associated URL links. This research addresses this issue by developing a Mention Detection Algorithm (MDA).

MDA operates by string matching the NEs and, on top of that, it also considers the Arabic morphological variations that can occur. For example, the actual linked NE (سعود الفيصل /sʕwd AlfySl/ 'Saud Alfaisal') can be successively mentioned as (الفيصل /AlfySl/ 'Alfaisal').

Subsequently, the automatically developed corpus was compiled, selecting only those sentences that had at least one NE. For the purposes of this study, a corpus size of more than 2 million tokens was compiled. This dataset was then freely distributed to the research community.

Chapter 5 also addresses the recognition that, in order to conduct a thorough experiment on fine-grained Arabic NER, a 'reasonable sized' gold standard manually created corpora across different genres must be developed. Therefore, the study compiled two corpora, namely NewsFANE$_{Gold}$ and WikiFANE$_{Gold}$, sized 170K and 500K respectively.

The creation of those corpora, including WikiFANE$_{Auto}$, facilitates the study of different characteristics of Arabic NEs within the textual context. Therefore, Chapter 5 concludes by conducting a comprehensive corpus-based comparative evaluation in order to study the density, length, structure and the fine-grained semantic distribution of NEs.

There are two important findings identified in Chapter 5. First, the automatic creation of fine-grained NE corpus by exploiting Wikipedia is manageable; this resource is considered promising for this purpose, taking into consideration the continually increasing number of articles. The second finding is that studying the nature, characteristics and behaviour of Arabic NEs from within a corpora yields better understanding towards developing an appropriate solution for NER.

Chapter 6 presents the development of the fine-grained NER in a pipeline structure, relying on supervised ML techniques. The pipeline structure consists of three components; the first component is the pre-processing, which involves normalising linguistic variations in the raw input text, including removing diacritics, if any, and tokenising the text using the scheme (conj+prep+suff)[1] provided by AMIRA. The second component is feature processing, which extracts informative features that will help the probabilistic model to perform the prediction. The third component is the probabilistic model, which is responsible for carrying out the learning step and then labelling the unseen text.

A baseline model is created based on Maximum Entropy (ME), by extracting the traditional lexical and contextual feature set. The reason for developing this baseline model is to have a comparable model to evaluate different methodologies. In this model, features are represented by relying solely on window-based representation, in other words n-gram representation, where the decision made for a token at position (i) is affected by the adjacent two tokens.

Chapter 6 also investigates the effect of developing another probabilistic model, a CRF classifier, and the injection of external knowledge, i.e. gazetteer, in relation to fine-grained NER. However, the traditional approach of encoding the annotation of NEs is to use a

[1]In this scheme the conjunctions, prepositions and suffixes are separated by white space.

BIO scheme, as seen in Table 6.6. This may not be the correct decision; therefore, different schemes, IO and BILOU, are examined in a controlled experiment.

The main findings of this chapter are:

1. The CRF probabilistic model outperforms ME in relation to fine-grained NER across different corpora.

2. Injecting external knowledge, i.e. a gazetteer, yields an improvement in performance.

3. The 'BILOU' scheme is proven to be the most suitable choice, rather than the traditional BIO scheme.

Chapter 7 investigates different methods of representing features. The most modern way of representing and involving features in the probabilistic model is by relying on window-based representation; this is sometimes called n-gram representation. However, the window-based representation has limitations in capturing informative features that distinguish between fine-grained semantic classes. Therefore, a new way of representing features is proposed, relying on the dependency structure of the sentence. This form of dependency representation enables the capturing of features beyond the scope of window-based representation. In addition, the integration of both dependency- and window-based representation as a hybrid representation is also investigated.

The main findings of this chapter are:

1. That dependency-based representation is a promising approach for fine-grained NER.

2. That dependency-based representation improves the classifier, allowing it to capture multi-word NEs, compared with a window-based representation.

3. That dependency-based representation boosts the performance of the probabilistic model, enabling it to properly classify the fine-grained NEs that share the same parent.

Thus far, the feature space is limited to window size, in window-based representations, or to sentence boundary, in dependency-based representations. Chapter 8 investigates an approach to extract global evidence that goes beyond those boundaries. Supporting the argument that similar words appear in similar contexts (Miller et al., 2004), the study utilises the richness of unannotated textual data by recruiting clustering techniques. In Chapter 8, the study relies on a hierarchical clustering algorithm called Brown's algorithm (Brown et al., 1992). Brown's algorithm works by maximising the mutual information between adjacent clusters; the hierarchical output is utilised and injected into the feature space. The findings show an improvement as a result of exploiting such features. The main finding of this chapter is that the hierarchical clustering technique is a suitable approach to exploiting global features in combination with both local and contextual features.

9.2 Main Contributions

To summarise the above discussion of the main thesis results, the present study has made the following key contributions:

1. The study examines the nature of Arabic NEs by exploring and defining their types and structures, and then conducting a corpus-based evaluation in order to study the characteristics and properties of NEs across corpora.

2. The study develops a methodology that exploits the richness of Arabic Wikipedia in order to automatically create a scalable fine-grained corpus and gazetteer. This result in:

   (a) WikiFANE$_{Auto}$: a fine-grained corpus of size 2M tokens

   (b) WikiFANE$_{Gazet}$: a fine-grained gazetteer comprises of 68355 entities

   In addition, the study develops two manually-created gold-standard fine-grained corpora from different genres and this result in:

   (a) NewsFANE$_{Gold}$: a newswire-based fine-grained corpus of size 170K tokens

(b) WikiFANE$_{Gold}$: a Wikipedia-based fine-grained corpus of size 500K tokens

3. The study develops fine-grained NER for Arabic by learning two probabilistic models (i.e. Maximum Entropy (ME) and Conditional Random Fields (CRF)) and investigates the effects of design decisions, including examining different encoding schemes and injecting external knowledge (i.e. gazetteer).

4. The study presents the development and the evaluation of a novel approach to representing features by relying on the dependency structure, which involves:

   (a) Identifying the limitations of the current window-based representation.

   (b) Utilising the dependency structure of the sentence, working towards achieving a dependency-based representation of the features.

5. The study exploits the unstructured textual data with the intention of developing and evaluating a hybrid-based approach to representing the features that capture global evidences, by performing word-level text clustering relying on Brown's (1992) hierarchical representation of clusters.

9.3 Future Work

Following this thesis, the future direction of the Arabic NER field could follow different paths, depending on the desired final goal. In this section, four directions that a researcher could take towards advancing research in the area of Arabic NER will be proposed.

1. The classification of fine-grained classes in one step is considered as an approach to developing fine-grained NER. Another alternative is to perform the classification in two stages. The first step is to classify into coarse-grained classes. The second step is to perform sub-classification for each coarse-grained class into their fine-grained classes. By this way, we could closely design the different set of features for each coarse-grained class to be involved in the classification process.

2. The investigation of the development of robust NER for Arabic is still a viable approach, particularly in the sense of engineering informative features. Including semantics into the feature space has not yet been investigated for Arabic NER. Lexical semantic knowledge resources, such as Arabic WordNet are a possible source of external knowledge. WordNet is a hierarchically organised lexical database that groups words into synsets, i.e. sets of near synonyms. In relation to NER, the exploitation of the synonym and hypernym hierarchies into the feature space is worth examining. For instance, synonyms could be beneficial features, particularly when the learning algorithm has the ability to capture synonyms for certain words in the training stage. Therefore, the same knowledge about the synonyms will be extremely valuable in helping to make decisions when testing unseen text. However, this is not a straightforward task, due to increased ambiguity, particularly when the number of words that share the same glyphs have different meaning depending on the context in which they appear. For example, the word "bank" has several meanings, depending on context, including a 'depository financial institution,' or 'sloping land'. WordNet does not perform a verification step through which to conclude the actual sense that fits the correct meaning of the context. Therefore, a word disambiguation stage is required in which to filter the WordNet results and only select those meanings that match the context.

3. Another possible future direction for developing Arabic NER is to address the problem of coreference resolution of NER within and across documents. For this task, the NE can be represented by name, nominal and pronominal mentions; for example, the NE (الملك عبدالله بن عبدالعزيز /Almlk ʕbdAllh bn ʕbdAlʕzyz/ 'King Abdullah bin Abdulaziz') can be expressed as (الملك عبدالله /Almlk ʕbdAllh/ 'King Abdullah'), (خادم الحرمين الشريفين /xAdm AlHrmyn Alšryfyn/ 'Custodian of the Two Holy

Mosques'), (هو /hw/ 'he') as by name, nominal and pronominal mentions respectively. Therefore, the task of NER is in seeking the ability to group those mentions that represent one object in the real word. This task becomes harder when applied over cross documents.

4. Despite the fact that upper level NLP applications, such as Question Answering (QA), require NER, there are several NER-related tasks that are equally important. Among those is the Relation Extraction (RE) task; RE is responsible for extracting the semantic relation between two or more NEs within a sentence or across sentences. The ultimate goal of RE is to convert the unstructured text into structured knowledge. This type of research requires, initially, making a decision regarding the type of relations that will be involved, as well as how to perform the extraction itself.

For all of the future directions mentioned above, the complexity of Arabic as a target language makes these tasks more challenging, which should encourage researchers to become involved in the project of developing ideas to solve these problems.

Appendix A
Locational and Personal Keywords

The full list of keywords attached to personal NEs is presented in Table A.1.

Table A.1: Full list of keywords attached to personal NEs

Arabic	**Transliteration**	**Gloss**
الملك	Almlk	King
الملكة	Almlkħ	Queen
الأمير	AlOmyr	Prince
الأميرة	AlOmyrħ	Princess
الوزير	Alwzyr	Minister
الوزيرة	Alwzyrħ	Secertary
الرئيس	Alrŷys	President
الرئيسة	Alrys	President (female)
النائب	AlnAb	Deputy
النائبة	AlnAb	Deputy (female)
السيد	Alsyd	Mr.
السيدة	Alsyd	Mrs. or Miss.
الشيخ	Alyx	Sheikh (Religious scholar)
العلامة	AllAm	Great scholar
العالم	AlAlm	Scholar
العالمة	AlAlm	Scholar (female)
المحافظ	AlmHAf	Conservative
المحافظة	AlmHAf	Conservative (female)
المدير	Almdyr	Manager
المديرة	Almdyr	Manger (female)

Continued on next page

Table A.1 – continued from previous page

Arabic	Transliteration	Gloss
الدكتور	Aldktwr	Dr.
الدكتورة	Aldktwr	Dr. (female)
المهندس	Almhnds	Eng.
المهندسة	Almhnds	Eng. (female)
الطبيب	AlTbyb	Doctor
الطبيبة	AlTbyb	Doctor (female)
البروفيسور	Albrwfyswr	Professor
البروفيسورة	Albrwfyswr	Professor (female)
الناخب	AlnAxb	Voter
الناخبة	AlnAxb	Voter(female)
الضابط	AlDAbT	Officer
الضابطة	AlDAbT	Officer (female)
العسكري	Alskry	Military officer
العسكرية	Alskry	Military officer (female)
القاضي	AlqADy	Judge
القاضية	AlqADy	Judge (female)
المحقق	AlmHqq	Detective
المحققة	AlmHqq	Detective (female)
المحامي	AlmHAmy	Lawyer
المحامية	AlmHAmy	Lawyer (female)
اللاعب	AllAb	Player
اللاعبة	AllAb	Player (female)
الرياضي	AlryADy	Athlete
الرياضية	AlryADy	Athlete (female)
المدرب	Almdrb	Coach
المدربة	Almdrb	Coach (female)
الفنان	AlfnAn	Artist
الفنانة	AlfnAn	Artist (female)
الصحفي	AlSHfy	Journalist
الصحفية	AlSHfy	Journalist (female)
السلطان	AlslTAn	Sultan

Continued on next page

Table A.1 – continued from previous page

Arabic	Transliteration	Gloss
الكاتب	AlkAtb	Writer
الكاتبة	AlkAtb	Writer (female)
المخرج	Almxrj	Producer
المخرجة	Almxrj	Producer (female)
المغني	Almny	Singer
المغنية	Almny	Singer (female)
القائد	AlqAd	Commander
القائدة	AlqAd	Commander (female)

The full list of keywords attached to locational NEs is presented in Table A.2.

Table A.2: Full list of keywords attached to locational NEs

Arabic	Transliteration	Gloss
مدينة	mdynħ	City
ولاية	wlAyħ	State
محافظة	mHAfǍDħ	Province
منطقة	mnTqħ	Region
بلدة	bldħ	Town
قرية	qryħ	Village
حي	Hy	District
نهر	nhr	River
مقاطعة	mqAT	Province
دولة	dwl	Country
ضاحية	DAHy	Suburb
مملكة	mmlk	Kingdom
جمهورية	jmhwry	Republic
امبراطورية	AmbrATwry	Empire
إقليم	Iqlym	Territory
إمارة	ImAr	Emirate
هجرة	hjr	Small town
مستعمرة	mstmr	Colony
بلد	bld	Country

Continued on next page

Table A.2 – continued from previous page

Arabic	Transliteration	Gloss
أرض	OrD	Land
شارع	Ar	Street
طريق	Tryq	Way
حقل	Hql	Field
ريف	ryf	Countryside

Appendix B
The Relation of Categories Used in this Thesis and those from ACE

The following table shows the relation of categories used in this thesis and those from ACE (2005). The symbols '+' and '*' represent categories added to and removed from ACE tagset respectively.

Table B.1: The Relation of Categories Used in this Thesis and those from ACE (2005)

Caorse-grained Classes	Fine-grained Classes
PER: Person^{+}	Politician^{+}, Athlete^{+}, Businessperson^{+}, Artist^{+}, Scientist^{+}, Police^{+}, Religious^{+}, Engineer^{+}, Group, Indeterminate*, Individual*.
ORG: Organisation	Government, Non-Governmental, Commercial, Educational, Media, Religious, Sports, Medical-Science, Entertainment.
LOC: Location	Address*, Boundary*, Water-Body, Celestial, Land-Region-Natural, Region-General*, Region-International*.
GPE: Geo-Political	Continent, Nation, State-or-Province, County-or-District, Population-Center, GPE-Cluster, Special*.
FAC: Facility	Building-Grounds, Subarea-Facility, Path, Airport, Plant.
VEH: Vehicle	Land, Air, Water, Subarea-Vehicle*, Underspecified*.
WEA: Weapon	Blunt, Exploding, Sharp, Chemical, Biological, Shooting, Projectile, Nuclear, Underspecified*.
PRO:Product^{+}	Book^{+}, Movie^{+}, Sound^{+}, Hardware^{+}, Software^{+}, Food^{+}, Drug^{+}.

List of References

Abdallah, S., Shaalan, K., and Shoaib, M. (2012). Integrating rule-based system with classification for Arabic named entity recognition. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, volume 7181 of *Lecture Notes in Computer Science*, pages 311–322. Springer Berlin Heidelberg.

AbdelRahman, S., Elarnaoty, M., Magdy, M., and Fahmy, A. (2010). Integrated machine learning techniques for Arabic named entity recognition. *IJCSI International Journal of Computer Science*, 7(4):27–36.

Abdul-Hamid, A. and Darwish, K. (2010). Simplified feature set for Arabic named entity recognition. In *Proceedings of the 2010 Named Entities Workshop*, pages 110–115, Uppsala, Sweden. Association for Computational Linguistics.

Abney, S. (2002). Bootstrapping. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 360–367. Association for Computational Linguistics.

Abuleil, S. (2004). Extracting names from Arabic text for question-answering systems. In *Proceedings of Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval (RIAO 2004)*, pages 638–647, Avignon, France.

ACE (2003). The Automatic Content Extraction 2003 (ACE03) evaluation plan. ftp://jaguar.ncsl.nist.gov/ace/doc/ace-evalplan-2003.v1.pdf [accessed 10 December 2013].

ACE (2004). The Automatic Content Extraction 2004 (ACE04) evaluation plan. http://www.itl.nist.gov/iad/mig/tests/ace/2004/doc/ace04-evalplan-v7.pdf [accessed 10 December 2013].

ACE (2005). The Automatic Content Extraction 2005 (ACE05) evaluation plan. http://www.itl.nist.gov/iad/mig/tests/ace/2005/doc/ace05-evalplan.v2a.pdf [accessed 10 December 2013].

Al-Jumaily, H., Martínez, P., Martínez-Fernández, J. L., and Van der Goot, E. (2012). A real time named entity recognition system for Arabic text mining. *Language resources and evaluation*, 46(4):543–563.

Al-Shalabi, R., Kanaan, G., Al-Sarayreh, B., Khanfer, K., Al-Ghonmein, A., Talhouni, H., and Al-Azazmeh, S. (2009). Proper noun extracting algorithm for Arabic language.

In *International Conference on IT to Celebrate S. Charmonman's 72nd Birthday*, page 28.128.9, Bangkok.

Alasiry, A., Levene, M., and Poulovassilis, A. (2012). Extraction and evaluation of candidate named entities in search engine queries. In Wang, X., Cruz, I., Delis, A., and Huang, G., editors, *Web Information Systems Engineering - WISE 2012*, volume 7651 of *Lecture Notes in Computer Science*, pages 483–496. Springer Berlin Heidelberg.

Alasiry, A., Levene, M., and Poulovassilis, A. (2014). Mining named entities from search engine query logs. In *Proceedings of the 18th International Database Engineering & Applications Symposium*, IDEAS '14, pages 46–56, New York, NY, USA. ACM.

Alfonseca, E. and Manandhar, S. (2002). An unsupervised method for general named entity recognition and automated concept discovery. In *Proceedings of the 1st International Conference on General WordNet, Mysore, India*, pages 34–43.

Algahtani, S. M. (2012). *Arabic Named Entity Recognition: A Corpus-Based Study*. PhD thesis, Computer Science, The University of Manchester.

Alias-i. (2008). LingPipe 4.1.0. http://alias-i.com/lingpipe [accessed 25 October 2014].

Alkhalifa, M. and Rodrguez, H. (2009). Automatically extending NE coverage of Arabic WordNet using Wikipedia. In *Proceedings of the 3rd International Conference on Arabic Language Processing CITALA2009*, Rabat, Morocco.

Allison, B., Guthrie, D., and Guthrie, L. (2006). Another look at the data sparsity problem. In Sojka, P., Kopeek, I., and Pala, K., editors, *Text, Speech and Dialogue*, volume 4188 of *Lecture Notes in Computer Science*, pages 327–334. Springer Berlin Heidelberg.

Althobaiti, M., Kruschwitz, U., and Poesio, M. (2013). A semi-supervised learning approach to Arabic named entity recognition. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 32–40, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

An, J., Lee, S., and Lee, G. G. (2003). Automatic acquisition of named entity tagged corpus from world wide web. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 165–168. Association for Computational Linguistics, Association for Computational Linguistics.

Asharef, M., Omar, N., and Albared (2012). Arabic named entity recognition in crime documents. *Journal of Theoretical and Applied Information Technology*, 44(1):1–6.

Attia, M., Toral, A., Tounsi, L., Monachini, M., and van Genabith, J. (2010). An automatically built named entity lexicon for Arabic. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC10)*, Valletta, Malta. European Language Resources Association (ELRA).

Avinesh, P. and Karthik, G. (2007). Part-of-speech tagging and chunking using conditional random fields and transformation based learning. In *Proceedings of the IJCAI 2007 Workshop On Shallow Parsing for South Asian Languages (SPSAL-2007)*, pages 21–24, Hyderabad, India.

Babych, B. and Hartley, A. (2003). Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other Language Technology Tools, Improving MT through other Language Technology Tools: Resources and Tools for Building MT*, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Balasuriya, D., Ringland, N., Nothman, J., Murphy, T., and Curran, J. R. (2009). Named entity recognition in Wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.

Beesley, K. (1996). Arabic finite-state morphological analysis and generation. In *Proceedings of the 16th Conference on Computational Linguistics*, pages 89–94, Copenhagen, Denmark. Association for Computational Linguistics.

Benajiba, Y., Diab, M., and Rosso, P. (2008a). Arabic named entity recognition: An SVM-based approach. In *Proceedings of 2008 Arab International Conference on Information Technology (ACIT)*, pages 16–18, Amman, Jordan. Association of Arab Universities.

Benajiba, Y., Diab, M., and Rosso, P. (2008b). Arabic named entity recognition using optimized feature sets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 284–293, Honolulu, Hawaii. Association for Computational Linguistics.

Benajiba, Y., Diab, M., and Rosso, P. (2009a). Arabic named entity recognition: A feature-driven study. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(5):926–934.

Benajiba, Y., Diab, M., and Rosso, P. (2009b). Using language independent and language specific features to enhance Arabic named entity recognition. *IEEE Transactions on Audio, Speech and Language Processing. Special Issue on Processing Morphologically Rich Languages*, 17(5):464–473.

Benajiba, Y. and Rosso, P. (2007). ANERsys 2.0: Conquering the NER task for the Arabic language by combining the maximum entropy with POS-tag information. In *Proceedings of the Workshop on Natural Language-Independent Engineering, 3rd Indian Int. Conf. on Artificial Intelligence, IICAI-2007*, page 18141823, Pune, India.

Benajiba, Y. and Rosso, P. (2008). Arabic named entity recognition using conditional random fields. In *Proceedings of the Workshop on HLT & NLP Within the Arabic World. Arabic Language and Local Languages Processing: Status Updates and Prospects, 6th International Conference on Language Resources and Evaluation*, pages 26–31, Marrakech, Morocco. LREC-2008.

Benajiba, Y., Rosso, P., and Benedíruiz, J. M. (2007). ANERsys: An Arabic named entity recognition system based on maximum entropy. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, volume 4394 of *Lecture Notes in Computer Science*, pages 143–153. Springer Berlin / Heidelberg.

Benajiba, Y. and Zitouni, I. (2009). Morphology-based segmentation combination for Arabic mention detection. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4):16:1–16:18.

Benajiba, Y., Zitouni, I., Diab, M., and Rosso, P. (2010). Arabic named entity recognition: Using features extracted from noisy data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 281–285, Uppsala, Sweden. Association for Computational Linguistics.

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., and Gauvain, J.-L. (2006). Neural probabilistic language models. In Holmes, D. and Jain, L., editors, *Innovations in Machine Learning*, volume 194 of *Studies in Fuzziness and Soft Computing*, pages 137–186. Springer Berlin Heidelberg.

Bidhendi, M. A., Minaei-Bidgoli, B., and Jouzi, H. (2012). Extracting person names from ancient islamic Arabic texts. In *Proceedings of Language Resources and Evaluation for Religious Texts (LRE-Rel) Workshop Programme, Eight International Conference on Language Resources and Evaluation (LREC 2012)*, pages 1–6.

Bikel, D. M., Schwartz, R., and Weischedel, R. M. (1999). An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media, Inc.

Bodnari, A., Deléger, L., Lavergne, T., Névéol, A., and Zweigenbaum, P. (2013). A supervised named-entity extraction system for medical text. In *Proceedings of the Conference and Labs of the Evaluation Forum (CLEF)*, Valencia, Spain.

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Brunstein, A. (2002). Annotation guidelines for answer types. http://www.ldc.upenn.edu/Catalog/docs/LDC2005T33/BBN-Types-Subtypes.html, LDC2005T33 [accessed 02 January 2012].

Buckwalter, T. (2002). Buckwalter Arabic morphological analyzer version 1.0. https://catalog.ldc.upenn.edu/LDC2002L49 [accessed 25 October 2014].

Carletta, J. (1996). Assessing agreement on classification tasks: the kappa statistic. *Computational linguistics*, 22(2):249–254.

Chinchor, N. and Robinson, P. (1997). MUC-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding*, Columbia, Maryland.

Cochran, W. G. (1950). The comparison of percentages in matched samples. *Biometrika*, pages 256–266.

Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 1–8. Association for Computational Linguistics.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Cunningham, H. (2002). GATE, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254.

Dakka, W. and Cucerzan, S. (2008). Augmenting Wikipedia with named entity tags. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*, pages 545–552, Hyderabad, India. Asian Federation of Natural Language Processing.

Dale, R. and Mazur, P. (2007). Handling conjunctions in named entities. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, volume 4394 of *Lecture Notes in Computer Science*, pages 131–142. Springer Berlin Heidelberg.

Darwish, K. (2002). Building a shallow Arabic morphological analyzer in one day. In *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*, pages 1–8, Philadelphia, Pennsylvania. Association for Computational Linguistics.

Darwish, K. (2013). Named entity recognition using cross-lingual resources: Arabic as an example. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1558–1567, Sofia, Bulgaria. Association for Computational Linguistics.

Darwish, K. and Gao, W. (2014). Simple effective microblog named entity recognition: Arabic as an example. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Darwish, K. and Magdy, W. (2014). *Arabic Information Retrieval*, volume 7. Now Publishers Inc.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science (JASIS)*, 41(6):391–407.

Desmet, B. and Hoste, V. (2014). Fine-grained dutch named entity recognition. *Language Resources and Evaluation*, 48(2):307–343.

Diab, M. (2009). Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking. In *2nd International Conference on Arabic Language Resources and Tools*, page 285288, Cairo.

Diab, M., Hacioglu, K., and Jurafsky, D. (2007). Automated methods for processing Arabic text: from tokenization to base phrase chunking. In *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Kluwer/Springer.

Diab, M., Moschitti, A., and Pighin, D. (2008). Semantic role labeling systems for Arabic using kernel methods. In *Proceedings of ACL-08: HLT*, pages 798–806, Columbus, United States.

Downey, D., Broadhead, M., and Etzioni, O. (2007). Locating complex named entities in web text. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 2733–2739, Hyderabad, India.

Ehrmann, M., Turchi, M., and Steinberger, R. (2011). Building a multilingual named entity-annotated corpus using annotation projection. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, pages 118–124, Hissar, Bulgaria.

Eiselt, A. and Figueroa, A. (2013). A two-step named entity recognizer for open-domain search queries. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 829–833, Nagoya, Japan. Asian Federation of Natural Language Processing.

Ekbal, A., Sourjikova, E., Frank, A., and Ponzetto, S. P. (2010). Assessing the challenge of fine-grained named entity recognition and classification. In *Proceedings of the 2010 Named Entities Workshop*, pages 93–101. Association for Computational Linguistics.

Elsebai, A. (2009). *A rules based system for named entity recognition in modern standard Arabic*. PhD thesis, University of Salford.

Elsebai, A. and Meziane, F. (2011). Extracting person names from Arabic newspapers. In *International Conference on Innovations in Information Technology (IIT)*, pages 87–89.

Elsebai, A., Meziane, F., and Belkredim, F. Z. (2009). A rule based persons names Arabic extraction system. *Communications of the International Business Information Management Association (IBIMA)*, 11(6):53–59.

Farber, B., Freitag, D., Habash, N., and Rambow, O. (2008). Improving NER in Arabic using a morphological tagger. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pages 2509–2514, Marrakech, Morocco. European Language Resources Association (ELRA).

Farghaly, A. and Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4):1–22.

Fehri, H., Haddar, K., and Ben Hamadou, A. (2011). Recognition and translation of Arabic named entities with NooJ using a new representation model. In *Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing*, pages 134–142. Association for Computational Linguistics.

Ferrández, S., Ferrández, O., Ferrández, A., and Muáoz, R. (2007). The importance of named entities in cross-lingual question answering. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria.

Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., and Dredze, M. (2010). Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 80–88. Association for Computational Linguistics.

Fleischman, M. (2001). Automated subcategorization of named entities. In *Proceedings of the Conference of the European Chapter of Association for Computational Linguistic*, pages 25–30, Toulouse, France. Association for Computational Linguistics.

Fleischman, M. and Hovy, E. (2002). Fine grained classification of named entities. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Taipei, Taiwan. Association for Computational Linguistics.

Florian, R., Ittycheriah, A., Jing, H., and Zhang, T. (2003). Named entity recognition through classifier combination. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL*, pages 168–171. Association for Computational Linguistics.

Fu, R., Qin, B., and Liu, T. (2011). Generating chinese named entity data from a parallel corpus. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 264–272, Chiang Mai, Thailand. Asian Federation of Natural Language Processing (AFNLP).

Giuliano, C. and Gliozz, A. (2008). Instance-based ontology population exploiting named-entity substitution. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 265–272, Manchester, United Kingdom. Association for Computational Linguistics.

Goweder, A. and De Roeck, A. (2001). Assessment of a significant Arabic corpus. In *Arabic NLP Workshop at ACL/EACL*, Toulouse,France.

Green, S. and Manning, C. D. (2010). Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 394–402. Association for Computational Linguistics.

Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: a brief history. In *Proceedings of the 16th Conference on Computational Linguistics*, pages 466–471, Copenhagen, Denmark. Association for Computational Linguistics.

Guo, J., Xu, G., Cheng, X., and Li, H. (2009). Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274. ACM.

Habash, N., Rambow, O., and Roth, R. (2009). Mada + tokan: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and

lemmatization. In *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, pages 102–109, Cairo, Egypt.

Habash, N. and Roth, R. M. (2009). Catib: The columbia Arabic treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224. Association for Computational Linguistics.

Habash, N., Soudi, A., and Buckwalter, T. (2007). On Arabic transliteration. In Soudi, A., Bosch, A. d., and Neumann, G., editors, *Arabic Computational Morphology*, volume 38 of *Text, Speech and Language Technology*, pages 15–22. Springer Netherlands.

Habash, N. Y. (2010). Introduction to Arabic natural language processing. In Hirst, G., editor, *Synthesis Lectures on Human Language Technologies*, volume 3, pages 1–187. Morgan & Claypool Publishers, 1 edition.

Hajic, J., Smrz, O., Zemanek, P., Pajas, P., Snaidauf, J., Beska, E., Kracmar, J., and Hassanova, K. (2004). Prague Arabic dependency treebank 1.0. https://catalog.ldc.upenn.edu/LDC2004T23 [accessed 15 October 2014].

Harabagiu, S., Moldovan, D., Clark, C., Bowden, M., Williams, J., and Bensley, J. (2003). Answer mining by combining extraction techniques with abductive reasoning. In *Proceedings of 12th Text Retrieval Conference (TREC)*, pages 375–382. National Institute of Standards & Technology (NIST).

He, X., Zemel, R. S., and Carreira-Perpindn, M. (2004). Multiscale conditional random fields for image labeling. In *Proceedings of the Computer vision and pattern recognition (CVPR)*, volume 2, pages II–695. IEEE.

Howe, J. (2006). The rise of crowdsourcing. *Wired magazine*, 14(6):1–4.

Hripcsak, G. and Rothschild, A. S. (2005). Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298.

Hsueh, P.-Y., Melville, P., and Sindhwani, V. (2009). Data quality from crowdsourcing: A study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, HLT '09, pages 27–35, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jurafsky, D. and Martin, J. H. (2000). *Speech & language processing*. Prentice Hall.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

Koulali, R. and Meziane, A. (2012). A contribution to Arabic named entity recognition. In *the 10th International Conference on ICT and Knowledge Engineering*, pages 46–52.

Kripke, S. (1972). *Naming and Necessity*, volume 40 of *Synthese Library*. Springer Netherlands.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*, pages 282–289, San Francisco, USA. Morgan Kaufmann Publishers Inc.

Lamar, M., Maron, Y., Johnson, M., and Bienenstock, E. (2010). SVD and clustering for unsupervised POS tagging. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 215–219. Association for Computational Linguistics.

Larkey, L. S., Ballesteros, L., and Connell, M. E. (2002). Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–282, Tampere, Finland. ACM.

Lavergne, T., Cappé, O., and Yvon, F. (2010). Practical very large scale CRFs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 504–513, Uppsala, Sweden. Association for Computational Linguistics.

Lease, M. (2011). On quality control and machine learning in crowdsourcing. In *Human Computation*.

Lee, C., Hwang, Y.-G., and Jang, M.-G. (2007). Fine-grained named entity recognition and relation extraction for question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 799–800. ACM.

Lee, C., Hwang, Y.-G., Oh, H.-J., Lim, S., Heo, J., Lee, C.-H., Kim, H.-J., Wang, J.-H., and Jang, M.-G. (2006). Fine-grained named entity recognition using conditional random fields for question answering. In Ng, H., Leong, M.-K., Kan, M.-Y., and Ji, D., editors, *Information Retrieval Technology*, volume 4182 of *Lecture Notes in Computer Science*, pages 581–587. Springer Berlin Heidelberg.

Lee, Y.-S., Papineni, K., Roukos, S., Emam, O., and Hassan, H. (2003). Language model based arabic word segmentation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 399–406, Sapporo, Japan. Association for Computational Linguistics.

Li, W., Li, J., Tian, Y., and Sui, Z. (2012). Fine-grained classification of named entities by fusing multi-features. In *Proceedings of COLING 2012: Posters*, pages 693–702, Mumbai, India. The COLING 2012 Organizing Committee.

Li, X. and Roth, D. (2006). Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(03):229–249.

Liang, P. (2005). Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

Lovins, J. B. (1968). *Development of a stemming algorithm.* MIT Information Processing Group, Electronic Systems Laboratory.

Maloney, J. and Niv, M. (1998). TAGARAB, a fast, accurate Arabic name recognizer using high-precision morphological analysis. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, pages 8–15, Montreal, Quebec, Canada. Association for Computational Linguistics.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

Martins, A. F., Smith, N. A., Xing, E. P., Aguiar, P. M., and Figueiredo, M. A. (2010). Turbo parsers: Dependency parsing by approximate variational inference. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 34–44. Association for Computational Linguistics.

Marton, Y., Habash, N., and Rambow, O. (2013). Dependency parsing of modern standard Arabic with lexical and inflectional features. *Computational Linguistics*, 39(1):161–194.

McCarthy, J. J. (1981). A prosodic theory of nonconcatenative morphology. *Linguistic inquiry*, 12:373–418.

McNamee, P., Snow, R., Schone, P., and Mayfield, J. (2008). Learning named entity hyponyms for question answering. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 799–804, Hyderabad, India. Asian Federation of Natural Language Processing (AFNLP).

Meftouh, K., Smaïli, K., Laskri, M. T., et al. (2008). Arabic statistical language modeling. In *9es Journées internationales d'Analyse statistique des Données Textuelles-JADT*, pages 837–844.

Mendes, A. C., Coheur, L., and Lobo, P. V. (2010). Named entity recognition in questions: Towards a golden collection. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta. European Language Resources Association (ELRA).

Mesfar, S. (2007). Named entity recognition for Arabic using syntactic grammars. In Kedad, Z., Lammari, N., Mtais, E., Meziane, F., and Rezgui, Y., editors, *Natural Language Processing and Information Systems*, volume 4592 of *Lecture Notes in Computer Science*, pages 305–316. Springer Berlin Heidelberg.

Miller, S., Guinness, J., and Zamanian, A. (2004). Name tagging with word clusters and discriminative training. In Susan Dumais, D. M. and Roukos, S., editors, *HLT-NAACL 2004: Main Proceedings*, pages 337–342, Boston, Massachusetts, USA. Association for Computational Linguistics.

Mnih, A. and Hinton, G. E. (2009). A scalable hierarchical distributed language model. *Advances in neural information processing systems (NISP)*, 21:1081–1088.

Mohammed, N. F. and Omar, N. (2012). Arabic named entity recognition using artificial neural network. *Journal of Computer Science*, 8(8):1285.

Mohit, B., Schneider, N., Bhowmick, R., Oflazer, K., and Smith, N. A. (2012). Recall-oriented learning of named entities in arabic wikipedia. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 162–173, Avignon, France. Association for Computational Linguistics.

Mollá, D., Zaanen, M., and Smith, D. (2006). Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop Sancta Sophia College*, pages 51–58, Sydney, Australia.

Morsi, A. and Rafea, A. (2013). Studying the impact of various features on the performance of conditional random field-based Arabic named entity recognition. In *Proceedings of the 10th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, pages 1–5, Ifrane, Morocco. IEEE.

Mostefa, D., Laïb, M., Chaudiron, S., Choukri, K., and Chalendar, G. (2009). A multilingual named entity corpus for Arabic, English and French. In Choukri, K. and Maegaard, B., editors, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt. The MEDAR Consortium.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Nadeau, D., Turney, P., and Matwin, S. (2006). Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In Lamontagne, L. and Marchand, M., editors, *Advances in Artificial Intelligence*, volume 4013 of *Lecture Notes in Computer Science*, pages 266–277. Springer Berlin Heidelberg.

Nezda, L., Hickl, A., Lehmann, J., and Fayyaz, S. (2006). What in the world is a Shahab? wide coverage named entity recognition for Arabic. In *Proceedings of the fifth conference on International Language Resources and Evaluation (LREC'06)*, pages 41–46, Genoa, Italy. European Language Resources Association (ELRA).

Ng, K., Tsai, F., Chen, L., and Goh, K. (2007). Novelty detection for text documents using named entity recognition. In *Proceedings of 6th International Conference on Information, Communications & Signal Processing (ICICS)*, pages 1–5, Singapore. IEEE.

Nobata, C., Sekine, S., Isahara, H., and Grishman, R. (2002). Summarization system integrated with named entity tagging and IE pattern discovery. In *Proceedings of the third conference on International Language Resources and Evaluation (LREC'02)*, pages 1742–1745, Spain. European Language Resources Association (ELRA).

Noguera, E., Toral, A., Llopis, F., and Muńoz, R. (2005). Reducing question answering input data using named entity recognition. In *Text, Speech and Dialogue*, volume 3658 of *Lecture Notes in Computer Science*, pages 428–434. Springer Berlin Heidelberg.

Nothman, J., Curran, J., and Murphy, T. (2008). Transforming Wikipedia into named entity training data. In *Proceedings of the Australian Language Technology Association Workshop*, pages 124–132, Hobart, Australia. ALTA.

Nothman, J., Murphy, T., and Curran, J. (2009). Analysing Wikipedia and gold-standard corpora for NER training. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 612–620, Athens, Greece. Association for Computational Linguistics.

Nouvel, D., Antoine, J.-Y., Friburger, N., and Soulet, A. (2012). Coupling knowledge-based and data-driven systems for named entity recognition. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, HYBRID '12, pages 69–77, Stroudsburg, PA, USA. Association for Computational Linguistics.

Parker, R., Graff, D., Chen, K., Kong, J., and Maeda, K. (2011). Arabic gigaword fifth edition [ldc2011t11]. http://catalog.ldc.upenn.edu/LDC2011T11 [accessed 20 December 2013].

Paşca, M. (2007). Weakly-supervised discovery of named entities using web search queries. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 683–690. ACM.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830.

Petasis, G., Cucchiarelli, A., Velardi, P., Paliouras, G., Karkaletsis, V., and Spyropoulos, C. (2000). Automatic adaptation of proper noun dictionaries through cooperation of machine learning and probabilistic methods. In *Proceedings of the 23rd ACM International SIGIR Conference on Research and Development in Information Retrieval*, pages 128–135, Athens, Greece.

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Rau, L. (1991). Extracting company names from text. In *Proceedings of the 7th IEEE Conference on Artificial Intelligence Applications*, pages 29–32, Miami Beach, USA. IEEE.

Richman, A. and Schon, P. (2008). Mining wiki resources for multilingual named entity recognition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1–9, Columbus, Ohio, USA. Association for Computational Linguistics.

Ryding, K. C. (2005). *A Reference Grammar of Modern Standard Arabic*. Cambridge University Press.

Sabou, M., Bontcheva, K., and Scharl, A. (2012). Crowdsourcing research opportunities: lessons from natural language processing. In *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies*, page 17. ACM.

Saleh, I., Darwish, K., and Fahmy, A. (2010). Classifying Wikipedia articles into NE's using SVM's with threshold adjustment. In *Proceedings of the 2010 Named Entities Workshop*, pages 85–92, Uppsala, Sweden. Association for Computational Linguistics.

Samy, D., Moreno, A., and Guirao, J. M. (2005). A proposal for an Arabic named entity tagger leveraging a parallel corpus. In *Proceedings of the Recent Advances in Natural Language Processing RANLP*, pages 459–465, Borovets, Bulgaria.

Sekine, S. and Nobat, C. (2004). Definition, dictionaries and tagger for extended named entity hierarchy. In *Proceedings of the 4th International Conference on Language Resources And Evaluation*, pages 1977–1980, Lisbon, Portugal. ELRA.

Sekine, S., Sudo, K., and Nobata, C. (2002). Extended named entity hierarchy. In *Proceedings of the third International Conference on Language Resources and Evaluation*, pages 1818–1824, Las Palmas, Spain. ELRA.

Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107. Association for Computational Linguistics.

Shaalan, K. (2005). Arabic gramcheck: A grammar checker for Arabic. *Software: Practice and Experience*, 35(7):643–665.

Shaalan, K. (2010). Rule-based approach in Arabic natural language processing. *the International Journal on Information and Communication Technologies (IJICT)*, 3(3):11–19.

Shaalan, K. (2013). A survey of Arabic named entity recognition and classification. *Computational Linguistics*, 40:469–510.

Shaalan, K. and Oudah, M. (2014). A hybrid approach to Arabic named entity recognition. *Journal of Information Science*, 40(1):67–87.

Shaalan, K. and Raza, H. (2007). Person name entity recognition for Arabic. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, pages 17–24, Prague, Czech Republic. Association for Computational Linguistics.

Shaalan, K. and Raza, H. (2008). Arabic named entity recognition from diverse text types. In Nordstrm, B. and Ranta, A., editors, *Advances in Natural Language Processing*, volume 5221 of *Lecture Notes in Computer Science*, pages 440–451. Springer Berlin Heidelberg.

Shaalan, K. and Raza, H. (2009). NERA: Named entity recognition for Arabic. *Journal of the American Society for Information Science and Technology*, 60:1652–1663.

Shihadeh, C. and Günter, N. (2012). ARNE-A tool for namend entity recognition from Arabic text. In *The Fourth Workshop on Computational Approaches to Arabic Script-based Languages*, pages 24–31, San Diego, USA.

Silberztein, M. (2005). NooJ: A linguistic annotation system for corpus processing. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, HLT-Demo '05, pages 10–11, Stroudsburg, PA, USA. Association for Computational Linguistics.

Stolcke, A. et al. (2002). SRILM-an extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing, ICSLP2002 - INTERSPEECH*, Denver, USA.

Taghva, K., Elkhoury, R., and Coombs, J. S. (2005). Arabic stemming without a root dictionary. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC)*, pages 152–157, Washington DC, USA.

Tardif, S., Curran, J., and Murphy, T. (2009). Improved text categorisation for Wikipedia named entities. In *Australasian Language Technology Association Workshop*, pages 104–108, Sydney, Australia.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL*, pages 142–147. Association for Computational Linguistics.

Tkachenko, M., Simanovsky, A., and Petersburg, S. (2012). Named entity recognition: Exploring features. In *Proceedings of the 11th Conference on Natural Language Processing (KONVENS)*, pages 118–127, Vienna, Austria.

Tkatchenko, M., Ulanov, A., and Simanovsky, A. (2011). Classifying Wikipedia entities into fine-grained classes. In *Proceedings of the 27th International Conference on Data Engineering Workshops (ICDEW)*, pages 212–217. IEEE.

Toutanova, K. and Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics*, pages 63–70. Association for Computational Linguistics.

Traboulsi, H. (2009). Arabic named entity extraction: A local grammar-based approach. In *Proceedings of the 2009 International Multiconference on Computer Science and Information Technology (IMCSIT 2009)*, pages 139–143, Mragowo, Polland.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics.

Voorhees, E. M. and Tice, D. M. (1999). The TREC-8 question answering track evaluation. In *TREC*.

Šuster, S. and Van Noord, G. (2014). From neighborhood to parenthood: the advantages of dependency representation over bigrams in Brown clustering. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1382–1391, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Xu, J., Fraser, A., and Weischedel, R. (2002). Empirical studies in strategies for Arabic retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 269–274. ACM.

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *The International Conference on Machine Learning (ICML)*, pages 412–420.

Zaghouani, W. (2012). RENAR: A rule-based Arabic named entity recognition system. *ACM Transactions on Asian Language Information Processing (TALIP)*, 11(1):2:12:13.

Zaghouani, W., Pouliquen, B., Ebrahim, M., and Steinberger, R. (2010). Adapting a resource-light highly multilingual named entity recognition system to Arabic. In *Proceedings of the 7th conference on International Language Resources and Evaluation (LREC'10)*, Malta. European Language Resources Association (ELRA).

Zayed, O. and El-Beltagy, S. (2012). Person name extraction from modern standard Arabic or colloquial text. In *Proceedings of the 8th International Conference on Informatics and Systems (INFOS)*, pages 44–48, Giza, Egypt.

Zhang, Z. (2013). *Named entity recognition: challenges in document annotation, gazetteer construction and disambiguation*. PhD thesis, The University of Sheffield.

Zirikly, A. and Diab, M. (2014). Named entity recognition system for dialectal arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 78–86, Doha, Qatar. Association for Computational Linguistics.

Zitouni, I. and Benajiba, Y. (2014). Aligned-parallel-corpora based semi-supervised learning for Arabic mention detection. *IEEE Transactions on Audio, Speech and Language Processing (TASLP)*, 22(2):314–324.

Zitouni, I. and Florian, R. (2009). Cross-language information propagation for Arabic mention detection. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4):17:1–17:21.

Zitouni, I., Sorensen, J., Luo, X., and Florian, R. (2005). The impact of morphological stemming on Arabic mention detection and coreference resolution. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 63–70. Association for Computational Linguistics.